

AD-782 841

**ROTORCRAFT FLIGHT SIMULATION WITH
AEROELASTIC ROTOR AND IMPROVED
AERODYNAMIC REPRESENTATIONS. VOLUME
III. PROGRAMMER'S MANUAL**

P. Y. Hsieh, et al

Bell Helicopter Company

Prepared for:

**Army Air Mobility Research and Development
Laboratory**

June 1974

DISTRIBUTED BY:

NTIS

**National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151**

**Best
Available
Copy**

EUSTIS DIRECTORATE POSITION STATEMENT

This report has been reviewed by the Eustis Directorate, U. S. Army Air Mobility Research and Development Laboratory and is considered to be technically sound.

The computer program resulting from this contract will be provided, upon request of qualified users, for use in the design and analysis of rotary-wing aircraft. Volume III of this report, a programmer's manual, has not been widely distributed, but will be provided with the computer program to aid in program installation.

The technical monitor for this contract was Mr. Edward E. Austin, Aeromechanics, Technology Applications Division.

ADDITIONAL	
NTIS	WPA 00130 <input checked="" type="checkbox"/>
DIC	EAT 00004 <input type="checkbox"/>
UNCLASSIFIED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	Avail. Rep./Special
A	

DISCLAIMERS

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission, to manufacture, use, or sell any patented invention that may in any way be related thereto.

Trade names cited in this report do not constitute an official endorsement or approval of the use of such commercial hardware or software.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed. Do not return it to the originator.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER USAAMRDL-TR-74-10C	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER AD-782841
4. TITLE (and Subtitle) ROTORCRAFT FLIGHT SIMULATION WITH AEROELASTIC ROTOR AND IMPROVED AERODYNAMIC REPRESENTATIONS, Volume III, Programmer's Manual		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report
7. AUTHOR(s) P. Y. Hsieh John M. Davis		6. PERFORMING ORG. REPORT NUMBER BHC Rpt. 299-099-815
8. PERFORMING ORGANIZATION NAME AND ADDRESS Bell Helicopter Company P. O. Box 482 Fort Worth, Texas 76101		9. CONTRACT OR GRANT NUMBER(s) DAAJ02-72-C-0098
11. CONTROLLING OFFICE NAME AND ADDRESS Eustis Directorate U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia 23604		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE June 1974
		13. NUMBER OF PAGES
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Reproduced by NATIONAL TECHNICAL INFORMATION SERVICE U S Department of Commerce Springfield VA 22151		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Aerodynamics Flight Simulation Dynamics Aeroelasticity Helicopters Structural Properties Rotors Control Computer Programs Numerical Analysis Digital Computers Rotary Wing Aircraft		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report consists of three volumes and documents the current version in the C81 family of rotorcraft flight simulation programs developed by Bell Helicopter Company. This current version of the digital computer program is referred to as AGAJ73. The new, revised, or alternate mathematical models incorporated into the program during the current contract are as follows:		

Block 20. Continued

- (1) Fuselage aerodynamic forces and moments (revised)
- (2) Aerodynamic surfaces (revised with two surfaces added)
- (3) External stores/aerodynamic brakes (new)
- (4) Rotor blade airfoil section distribution (new)
- (5) Rotor-induced velocity distribution (alternate)
- (6) Rotor unsteady aerodynamics (alternate)
- (7) Rotor wake effect at aerodynamic surfaces (alternate)
- (8) Method for numerically integrating rotorcraft equations of motion (alternate)

This volume, the Programmer's Manual, contains the information necessary to set up and support the computer program. Specifically, it includes cross-references of FORTRAN COMMON BLOCK variables, a catalog of subroutines, and a discussion of programming considerations. The listings and related software for the computer program documented in this report are unpublished data which are on file at the Eustis Directorate, U. S. Army Air Mobility Research and Development Laboratory (USAAMRDL), Fort Eustis, Virginia. Volume I, The Engineer's Manual, documents the background and development of the current version of the program. Volume II, the User's Manual, contains the detailed information necessary for setting up an input data deck and interpreting the computed data.

ia

PREFACE

This report and its accompanying computer program were developed under Contract DAAJ02-72-C-0098 awarded in June 1972 by the Eustis Directorate of the U. S. Army Air Mobility Research and Development Laboratory (USAAMRDL). In addition to the work performed under this contract, the report and computer program include the documentation and program features developed under USAAMRDL Contracts DAAJ02-70-C-0063 and DAAJ02-73-C-0086. The contractor and USAAMRDL have agreed that the computer program documented herein is the new master version of the program. Hence, this report supercedes all previous versions of the C81 program and documentation.

Technical program direction was provided by Mr. E. E. Austin of USAAMRDL. Principal Bell Helicopter personnel associated with the current contract were Messrs. B. L. Blankenship, J. M. Davis, and P. Y. Hsieh, and Dr. B. T. Waak. In addition, Dr. R. L. Bennett and Mr. B. J. Bird assisted in coordinating the work and documentation prepared under the two previous contracts noted above with that prepared under this contract.

TABLE OF CONTENTS

	<u>Page</u>
PREFACE	iii
LIST OF ILLUSTRATIONS	vi
LIST OF TABLES	vi
1. INTRODUCTION	1-1
2. OPERATING ENVIRONMENT AND PROCEDURES	2-1
3. GENERAL PROGRAMMING AIDS	3-1
3.1 Marco Flow Charts	3-1
3.2 Fortran Subroutines in AGAJ73	3-1
3.3 Assembly Language Subprograms in AGAJ73.	3-10
3.4 FORTRAN Subroutines in GDAJ07	3-12
3.5 Labeled Commons in AGAJ73	3-13
3.6 Labeled Commons in GDAJ07	3-15
4. DETAILED PROGRAMMING AIDS	4-1
4.1 Control Section Cross-Reference	4-1
4.2 Program Listings	4-1
4.3 Programming Option in Numerical Integration	4-2
4.4 Reduction of Program Memory Requirements	4-2
4.5 Switch for Diagnostic Data From STAB.	4-4

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
2-1	Job Control Language To Run AGAJ73 and GDAJ07	2-9
3-1	Flow Chart of System Structure	3-16
3-2	Flow Chart of Trim Process	3-17
3-3	Flow Chart of Stability Analysis	3-18
3-4	Flow Chart of Maneuver With Runge-Kutta Method	3-19
3-5	Flow Chart of Maneuver With Hamming's Method	3-20
3-6	Flow Chart of GDAJ07.	3-21
4-1	Job Control Language To Run AS812A Data	4-32

LIST OF TABLES

<u>Table</u>		<u>Page</u>
2-1	Linkage Editor Cards for OVERLAY of AGAJ73	2-3
2-2	Linkage Editor Data Cards for HIARCHY Support of AGAJ73	2-5
2-3	Linkage Editor Data Cards for OVERLAY of GDAJ07.	2-6
2-4	Input/Output Units Used in AGAJ73	2-7
4-1	Control Section Cross-Reference for AGAJ73	4-5
4-2	Cross-Reference of Multiple Entry Subroutines in AGAJ73	4-25
4-3	Control Section Cross-Reference for GDAJ07	4-26
4-4	Reduction of AGAJ73 Storage Requirements	4-30
4-5	STAB Diagnostic Switch in AGAJ73.	4-31

1. INTRODUCTION

This manual documents the rotorcraft flight simulation program, designated AGAJ73, and its post processor for data reduction, designated GDAJ07. To the user, this system appears as a single program; to the programmer, the two programs are very different. This documentation is for the programs as they were written for, and are being used on, an IBM System/360 Model 65 Computer at Bell Helicopter Company.

The information in this volume is of two types. Section 2 contains the information necessary to get the programs operational on a computer compatible with the installation at Bell Helicopter Company. If the programs are to be modified in any way, the programmer will need the information in Sections 3 and 4 of this volume.

2. OPERATING ENVIRONMENT AND PROCEDURES

The System/360 environment under which this program system is maintained is Operating System/360 Option IV (MVT). Input on the system reader is controlled by Houston Automatic Spooling Program (HASP) II, as is system output destined for on-line printer or card punch. As delivered under this contract, the program requires approximately a 1000K region of memory (1K = 1024 bytes, 1000K \approx 1 megabyte). Methods for reducing this large memory requirement are discussed below and in Section 4.4. The Basic Sequential Access Method routines are retained in a resident LINKPACK area. Scratch data sets are directed to a Telex 5312 Direct Access Storage Facility. Tape data sets are recorded on Telex 6420 Tape Drives. The recording format for the CALCOMP PLOTTAPE is 9 track, 800 bpi, NRZI.

All other tape data sets are recorded as 9 track, 1600 fci phase encoded. To run the program under OS/360, at least 262,144 (256K) bytes of main memory are required for operation with Option I (PCP) or Option II (MFT) systems. At least 524,288 (512K) bytes of main memory are required for Option IV (MVT).

The program has been maintained on the IBM System/360 FORTRAN H Compiler in USA FORTRAN. Compiler options used are SOURCE, EBCDIC, NOLIST, NODECK, LOAD, MAP, NOEDIT, ID, XREF, and OPT=2. Since the FORTRAN H Compiler performs essential optimization functions, compilation of this program under any other compiler or using optimization level less than "2" on the H Compiler will result in degraded performance in speed. Some FORTRAN language features peculiar to IBM FORTRAN definitions have been noted explicitly in individual routines.

AGAJ73 can be link-edited in several different ways. For example, with the OVERLAY structure shown in Table 2-1, the entire program can be loaded into either main or extended memory; with the HIARCHY support structure shown in Table 2-2, part of the program can be loaded into main memory and the rest into extended memory; or if neither OVERLAY nor HIARCHY structure is used, the entire program can be loaded into either main or extended memories. The best way to link-edit this program depends upon the facilities available at the local installation.

Although this program is quite large, it can be run in less than 300K if the OVERLAY structure shown in Table 2-1 is used and the size of several large arrays is reduced according to the procedure outlined in Section 4.4. If the region size available is greater than 300K, the OVERLAY shown in Table 2-1 can be relaxed to reduce the overhead time associated with OVERLAY. By using the HIARCHY support structure shown in Table 2-2 and also following the procedures in Section 4.4, the program can run in a 206K region of main memory and a 450K region of extended memory. It would need approximately another 300K if the reduction of array sizes mentioned above is not done. Since the buffers are in extended memory, the amount of extended memory used depends upon the units opened.

The input data for the linkage editor for GDAJ07 are shown in Table 2-3. This OVERLAY structure results in a program whose longest segment is 142K bytes. After a buffer allocation, the size of the region needed is 184K bytes.

The Job Control Linkage (JCL) used to run a typical set of data is shown in Figure 2-1. If other than HIARCHY structure is used, the REGION card in the first step should be changed accordingly from that shown in Figure 2-4. Tables 2-4 and 2-5 give the unit type, what it is used for, and which subroutines refer to it for AGAJ73 and GDAJ07, respectively.

The JCL has been prepared such that the job is in two steps: C81 and C81PLOT. The C81 step executes AGAJ73, and the C81PLOT step executes GDAJ07 even if the first step abends. This was done so that if a maneuver abends on the first step, the user will still get plots of what was run.

TABLE 2-1. LINKAGE EDITOR CARDS FOR OVERLAY OF AGAJ73

```

OVERLAY ALPHA  INITIALIZATION SEGMENT
INSERT  POSRED,REDID,REDRWK,REDSWK,START,WKTABN,IHCNAMEL
OVERLAY BETA
INSERT  FFRCHK,JSTRED,LGCINT,LIZE,NPUTOT,RBOY,READIN
INSERT  REDATB,REDBMS,REDCL,XSTINT,ZERO
OVERLAY BETA
INSERT  FUSINT,INBLD,INRU,INRTR,INSCAS,JFBGIN,MODAL,PYLINT
INSERT  RTINIT,STBZIN,WPM DAL,XCONIN
OVERLAY BETA
INSERT  MNEM,TABFIX,TABOUT,TURN,YRINIT,YSINIT
OVERLAY ALPHA  HARMONIC ANALYSIS
INSERT  HARM,LOADT
OVERLAY ALPHA  STABILITY ANALYSIS SEGMENT
INSERT  ALLMAT,ALSTAB,INVERS,IOMAT,MORDRS,MODES,PUNCH,WRMS
INSERT  IHCCLABS,IHCCLAS,IHCCLSQ,IHCCSABS,IHCCSAS,IHCFPT
INSERT  IHCLSQRT
OVERLAY ALPHA  GENERAL PURPOSE SEGMENT
INSERT  ANAL,ANDOT,AZMINT,AZMUTH,CDCL,CLCD,CMCALC,DIFFER
INSERT  DOTX,FOCUS,FORMC,FUSFNM,HRESP,INTERQ,ITROT,MBAL
INSERT  NSTED,RADBN,RADIAL,RADOUT,RGUST,ROTAN,RTWAKE,SOLVE
INSERT  STBWAK,STBZFM,SWSRAT,TABINT,TRMANU,UNSTED,WING,WRFM
INSERT  WRTMNV,WSHDUF,XSTORE,IHCSASCN
OVERLAY BETA
INSERT  AJACOB,JACOB,DAMPER,WRVP
OVERLAY GAMMA  STABILITY ANALYSIS SEGMENT
INSERT  WRINST
OVERLAY DELTA
INSERT  INSTAB
OVERLAY DELTA
INSERT  STAB,WRSTAB
OVERLAY GAMMA  TRIM SEGMENT
INSERT  TRIM,TVT
OVERLAY DELTA
INSERT  RPTPG
OVERLAY DELTA
INSERT  TRM1
OVERLAY DELTA
INSERT  WRTRIM
OVERLAY DELTA
INSERT  ITRIM,PDZ1

```

TABLE 2-1. Concluded.

OVERLAY BETA MANEUVER SEGMENT
 INSERT DERIV, HAMPS, HPCG, MANU, MANUV, QUAN, RINGKTA, SAVE IC, WAG
 OVERLAY GAMMA
 INSERT SCASIT
 OVERLAY GAMMA
 INSERT VARI
 OVERLAY DELTA
 INSERT VCNTKL
 OVERLAY DELTA
 INSERT SUPERP
 OVERLAY DELTA
 INSERT MTLT
 OVERLAY DELTA
 INSERT GUST
 OVERLAY DELTA
 INSERT EXTORS
 OVERLAY DELTA
 INSERT CNTM
 OVERLAY DELTA
 INSERT MOMB
 OVERLAY ONE(REGION)
 INSERT INIT, RFSTRT, SIVAR, TIMLP, TIVAR, WRMANU
 OVERLAY ONE
 INSERT CONSTR, STBD
 ENTRY MAIN

TABLE 2-2. LINKAGE EDITOR DATA CARDS FOR HIARCHY SUPPORT OF AGAJ73

HIARCHY 1, INCECONH2	HIARCHY 1, INSTAR	HIARCHY 1, SCASIT
HIARCHY 1, INCECONH	HIARCHY 1, INVERS	HIARCHY 1, SIVAR
HIARCHY 1, INCEFIOS	HIARCHY 1, IOMAT	HIARCHY 1, STAMAN
HIARCHY 1, INCEFNTH	HIARCHY 1, JACOBI	HIARCHY 1, STARAN
HIARCHY 1, INCERRM	HIARCHY 1, JFSGIN	HIARCHY 1, START
HIARCHY 1, INCETRCH	HIARCHY 1, JSTRED	HIARCHY 1, STONAK
HIARCHY 1, INCFCVTH	HIARCHY 1, LGCINT	HIARCHY 1, STBZIN
HIARCHY 1, INCPOPT	HIARCHY 1, LIZE	HIARCHY 1, SUPERP
HIARCHY 1, INCNAMEL	HIARCHY 1, LOADT	HIARCHY 1, SMSRAT
HIARCHY 1, INCUATBL	HIARCHY 1, MAIN	HIARCHY 1, TABFIX
HIARCHY 1, INCUOPT	HIARCHY 1, MANU	HIARCHY 1, TABOUT
HIARCHY 1, ABDUMP	HIARCHY 1, MANUV	HIARCHY 1, TILT
HIARCHY 1, AJACOB	HIARCHY 1, MORDRS	HIARCHY 1, TIMLP
HIARCHY 1, ALSTAB	HIARCHY 1, MNEN	HIARCHY 1, TIVAR
HIARCHY 1, CGZARN	HIARCHY 1, MODAL	HIARCHY 1, TOPLOT
HIARCHY 1, CNTH	HIARCHY 1, MODES	HIARCHY 1, TRIM
HIARCHY 1, CONSTB	HIARCHY 1, MONB	HIARCHY 1, TRMANU
HIARCHY 1, CONTRM	HIARCHY 1, NTLT	HIARCHY 1, TRM1
HIARCHY 1, DATE	HIARCHY 1, NOPS	HIARCHY 1, TURN
HIARCHY 1, DAMPER	HIARCHY 1, NPUTOT	HIARCHY 1, VARI
HIARCHY 1, DERIV	HIARCHY 1, PDSRED	HIARCHY 1, VCNTRL
HIARCHY 1, ERRCHK	HIARCHY 1, PDZ1	HIARCHY 1, WAG
HIARCHY 1, EXTORS	HIARCHY 1, PUNCH	HIARCHY 1, WKTABN
HIARCHY 1, FOCUS	HIARCHY 1, PYLINT	HIARCHY 1, WRFH
HIARCHY 1, FORMK	HIARCHY 1, QUAN	HIARCHY 1, WRINST
HIARCHY 1, FORMK1	HIARCHY 1, RADOUT	HIARCHY 1, WRMANU
HIARCHY 1, FORVYV	HIARCHY 1, RBDY	HIARCHY 1, WRMDAL
HIARCHY 1, FORVY1	HIARCHY 1, READIN	HIARCHY 1, WRNS
HIARCHY 1, FOSWK	HIARCHY 1, REDATJ	HIARCHY 1, WROT1
HIARCHY 1, FOSWK1	HIARCHY 1, RFDONS	HIARCHY 1, WRSTAB
HIARCHY 1, FUSINT	HIARCHY 1, REDCL	HIARCHY 1, WRTMNV
HIARCHY 1, GUST	HIARCHY 1, REDID	HIARCHY 1, WRTRIM
HIARCHY 1, HAHNS	HIARCHY 1, REDRWK	HIARCHY 1, WRVP
HIARCHY 1, HPCG	HIARCHY 1, REDSWK	HIARCHY 1, XCONIN
HIARCHY 1, INBLD	HIARCHY 1, RESTRT	HIARCHY 1, XSTINT
HIARCHY 1, INIT	HIARCHY 1, RNKTA	HIARCHY 1, YRINIT
HIARCHY 1, INRO	HIARCHY 1, RPTPG	HIARCHY 1, YSINIT
HIARCHY 1, INRTR	HIARCHY 1, RTINIT	HIARCHY 1, ZERO
HIARCHY 1, INSCAS	HIARCHY 1, SAVEIC	HIARCHY 1, ZLLCAL
HIARCHY 1, INSTAB	HIARCHY 1, SAVOLD	ENTRY MAIN

TABLE 2-3. LINKAGE EDITOR DATA CARDS FOR OVERLAY OF GDAJ07

IEW0000	OVERLAY ALPHA
IFW0000	INSERT CURVET
IEW0000	INSERT IHCLATN2
IFW0000	INSERT IHCLSORT
IEW0000	INSERT CALL
IEW0000	INSERT TIMPTS
IEW0000	OVERLAY ALPHA
IEW0000	INSERT CALCOM
IEW0000	INSERT IHCFRXP1
IEW0000	INSERT LINE
IEW0000	INSERT NEXTTIME
IFW0000	INSERT NUMBER
IFW0000	INSERT SYMBOL
IEW0000	OVERLAY BETA
IEW0000	INSERT FSFT
IEW0000	INSERT HARM
IEW0000	INSERT IHCSORT
IEW0000	INSERT PLOTFR
IEW0000	INSERT AXIS#
IEW0000	INSERT IHCFRXP2
IEW0000	INSERT IHCLSCN
IEW0000	INSERT IHCEXP
IEW0000	INSERT IHCSLOG
IEW0000	INSERT SCALE#
IEW0000	OVERLAY BETA
IEW0000	INSERT SCALIT
IEW0000	INSERT CALC81
IEW0000	INSERT PPLOT
IEW0000	INSERT SCLFIX
IEW0000	INSERT INPLOT
IEW0000	ENTRY MAIN

TABLE 2-4. INPUT/OUTPUT UNITS USED IN AGAJ73

Unit No.	Type	Used for	Used by Subroutine
1	direct access	Permanent data storage in partitioned data set (PDS)	JSTRED, REDATB, REDBMS, REDCL, REDID, REDRWK, REDSWK
2	tape	New restart tape	RESTRT
3	direct access	Utility storage of time-variant trim flapping history and storing maneuver time history to pass to GDAJ07	INIT, LOADT, MAIN, MANU, MANUV, RESTRT, TVT
4	tape	Old restart tape	RESTRT
5	card reader	Input data	MAIN
6	printer	Printed output	ALSTAB, AZMUTH, CDCL, CLCD, DAMPER, ERRCHK, EXTORS, FUSINT, HAMMS, HPCG, INRO, INSTAB, IOMAT, ITRIM, ITROT, JFBCIN, LIZE, LOADT, MAIN, MBAL, MNEM, NPUTOT, RADOUT, READIN, REDID, REDRWK, RPTPG, SIVAR, STAB, TABOUT, TIVAR, TRIM, TRM1, TURN, VIND, WAG, WRFM, WRINST, WRMANU, WRMDAL, WRMS, WROT1, WRSTAB, WRTRIM, WRTRIM, WRVP, XCONIN, YRINIT, YSINIT
7	card punch	Punched output	PUNCH
10	direct access	Utility storage of AGAJ73 input data	JSTRED, MAIN, READIN, REDATB, REDBMS, REDCL, REDID, REDRWK, REDSWK
11	direct access	Passing input data to GDAJ07	MAIN, READIN

TABLE 2-4. Continued.			
Unit No.	Type	Used for	Used by Subroutine
14	direct access	Stored and retrieved initial conditions when Hamming's method is used	SAVEIC

TABLE 2-5. INPUT/OUTPUT UNITS USED IN GDAJ07			
Unit No.	Type	Used for	Used by Subroutines
3	direct access	Maneuver time history from AGAJ73 or Tape 8	CURVET, C81L, FSFT, MAIN, SCALIT
6	printer	Printed output	CALC81, CURVET, C18L, FSFT, PLOT, WROT1
8	tape	Old time history tape	C81L
9	tape	New time history tape	C81L
10	direct access	Input data from AGAJ73	CURVET, C81L, FSFT, MAIN, SCALIT
PLOT-TAPE	tape	Plot maneuver time history in GDAJ07	PLOT8R

```

//E2712370 JOB (AGAJ7300,638,69109304,DP20,T,02),IPY          2841*,*
//
//MSOLEVFL=1,CLASS=C
//C01PRDC PRDC  PROG=AGAJ7300,LIB=ENGTEST,P3SP=100,BLK=7292,      000000010
//RESTART=RESTO=NULLFILE,RESTO=NULLFILE,THIN=NULLFILE,          000000020
//THINIT=NULLFILE,THIN=0,SYSPLOT=NULLFILE,                        000000030
//GRAPH=GDAJ07,LIB1=ENGTEST                                       000000040
//                                                                    000000050
//PARAMETERS ON THE EXEC STATEMENT:                                000000060
//NAME      DEFAULT      USAGE                                     000000070
//-----
//PROG      AGAJ73        PROGRAM NAME                             000000080
//LIB        PNRH.PROD1    LIBRARY WHERE PROGRAM RESIDES           000000090
//P3SP       100          NO. OF CYLINDERS FOR PTO3P001            000000100
//BLK        7292        BLOCKSIZE OF PTO3P001, PTO3P001, AND     000000110
//                                PTO3P001                          000000120
//RESTART    NULLFILE     OSNAME FOR RESTART TAPE INPUT            000000130
//RESTO      NULLFILE     OSNAME FOR RESTART TAPE OUTPUT          000000140
//THIN       NULLFILE     OSNAME FOR TIME HISTORY INPUT            000000150
//THINIT     NULLFILE     OSNAME FOR TIME HISTORY OUTPUT          000000160
//THIN       0            VOL=SER FOR TIME HISTORY INPUT TAPE     000000170
//SYSUT1     (TEMP)       OSNAME FOR PTO3P001                     000000180
//TPS        ANY TAPE DRIVE                                       000000190
//DOT        ANY DUAL DENSITY TAPE DRIVE                          000000200
//SYSPLOT    NULLFILE     OSNAME FOR PLOT TAPE                     000000210
//                                                                    000000220
//                                                                    000000230
//C01 EXEC PGM=CPRD,REGION=(204K,550K)                             000000240
//STEP110 DD DISP=SHR,DSN=CL10                                     000000250
//PT01P001 DD UNIT=2314,DSN=COL1072(DUM),VOL=SER=JOBLOG,DISP=SHR, 000000260
//DCB=HARCHY=1,LABEL=(,.,IN)                                       000000270
//DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)                             000000280
//PT02P001 DD UNIT=(TPS,,OFFER),DISP=(,CATLG,DELETE),DSN=CRESTO, 000000290
//DCB=HARCHY=1                                                     000000300
//PT03P001 DD UNIT=SYS0A,SPACE=(CYL,(P3SP)),DSN=ESYSUT1,          000000310
//DCB=(RECFM=VBS,LRECL=3644,BLKSIZE=60K,HARCHY=1),              000000320
//DISP=(INL,PASS)                                                  000000330
//PT04P001 DD UNIT=(TPS,,OFFPR),DISP=OLD,DSN=CRESTI,              000000340
//DCB=HARCHY=1                                                     000000350
//PT05P001 DD NMAMP=IN                                             000000360
//PT06P001 DD SYSINUT=A                                           000000370
//PT07P001 DD SYSOUT=B                                             000000380
//PT10P001 DD UNIT=SYS0A,SPACE=(TRK,(10)),DSN=ESYSIN1,            000000390
//DCB=(RECFM=FB,LRECL=80,BLKSIZE=3600,HARCHY=1),                 000000400
//DISP=(INL,PASS)                                                  000000410
//PT11P001 DD DSN=ESYSIN2,UNIT=SYS0A,SPACE=(TRK,(10)),           000000420
//DCB=(RECFM=FB,BLKSIZE=80,HARCHY=1),DISP=(INL,PASS)             000000430
//PT14P001 DD UNIT=SYS0A,SPACE=(TRK,(8,4)),DISP=(INL,DELETE),    000000440
//DCB=HARCHY=1                                                     000000450
//C01PLOT EXEC PGM=CGRAPH,COND=(14,LF,C01),EVEN)                 000000460
//STEP110 DD DISP=SHR,DSN=CL101                                     000000470
//PT03P001 DD DISP=(INL,DELETE),DSN=ESYSUT1,DCB=HARCHY=1         000000480
//PT06P001 DD SYSOUT=A                                             000000490
//PT08P001 DD UNIT=(TPS,,OFFPR),DISP=OLD,DSN=ETHIN,VOL=SER=ETHSR, 000000500
//DCB=HARCHY=1                                                     000000510
//PT09P001 DD UNIT=(TPS,,OFFER),DISP=(,KEEP,DELETE),DSN=ETHOUT,  000000520
//DCB=(RECFM=VBS,LRECL=3644,BLKSIZE=60K,HARCHY=1),              000000530
//PT10P001 DD DISP=(INL,DELETE),DSN=ESYSIN2,DCB=HARCHY=1         000000540
//PLOT TAPE DD UNIT=(DOT,,OFFPR),DSN=ESYSLOT,LADFL=(,NL),VOL=PRIVATE, 000000550
//DCB=HARCHY=1                                                     000000560
//PPM THIS IS THE END OF THE IN-STREAM PROCEDURE                 000000570
//EXEC C01PRDC
//IN DD *

```

Reproduced from
best available copy.

*REGION size shown in for use with HARCHY support. If OVERLAY is used or entire program is run only in main or only in extended memory, the REGION size must be changed accordingly.

Figure 2-1. Job Control Language To Run AGAJ73 and GDAJ07 Data.

3. GENERAL PROGRAMMING AIDS

3.1 MARCO FLOW CHARTS

The flow charts in Figures 3-1 through 3-6 describe the functional structure of the program without regard to flow by subroutine. Figure 3-1 shows the total program structure and is a composite of Figures 1-1 through 1-13 in Volume II. Figure 3-2 provides some detail of the trim process in Figure 3-1. Figure 3-3 amplifies the stability analysis shown in Figure 3-1. Figures 3-4 and 3-5 give some flow logics of Runge-Kutta Method and Hamming's Predictor-Corrector Method employed, respectively, in maneuver functions in Figure 3-1. Figure 3-6 details the data reduction program.

3.2 FORTRAN SUBROUTINES IN AGAJ73

The FORTRAN subroutines contained in AGAJ73 are listed in alphabetical order including the main program, which is called MAIN. A few remarks are made for each subroutine which indicate its general purpose or use in the program. In the case of subroutines with multiple entry points, the names of all the entries are given after the name of the subroutine in the order that they occur in the subroutine.

(1) AJACOB, AJACB1. This subroutine handles computation of quantities which depend upon variables which are changed in either subroutine TRIM or subroutine STAB in order to compute partial derivatives. These quantities are then calculated and used in the computation of forces and moments.

(2) ALLMAT, ALLVEC. The stability analysis uses this subroutine to compute eigenvalues and eigenvectors.

(3) ALSTAB. This subroutine calls ALLMAT and processes and prints out the results of the stability analysis.

(4) ANAL. Output of this subroutine consists of the total summation of forces and moments.

(5) AZMINT. It initializes some variables which will be used by subroutine AZMUTH.

(6) AZMUTH. This subroutine in the rotor analysis does the calculation and integration at each blade azimuth position.

(7) CDCL. This subroutine uses the local angle of attack and Mach number plus the airfoil aerodynamic inputs to compute the steady-state lift, drag, and pitching moment coefficients for each blade segment of the rotor.

(8) CGZARM, CGYARM, CGXARM. This subroutine calculates moment arms about the cg for the aerodynamic surfaces whenever cg is shifted.

- (9) CLCD. This subroutine is similar to subroutine CDCL except that CLCD computes the three aerodynamic coefficients for the wing and stabilizing surfaces.
- (10) CMC/LC. This subroutine interpolates on the Carta tables to produce the contribution of unsteady aerodynamics to the pitching moment. It is the major section of the BUNS unsteady aerodynamic model.
- (11) CNTM, AUXJ, FLAP. Some of the forcing functions in a maneuver may be timed to start after the rotors have been stopped. This subroutine converts those relative times to absolute times. It also varies jet thrust and RPM dependent flapping stops during a maneuver.
- (12) CONSTB. This is the control program for the stability analysis.
- (13) CONTRM. This is the control program of the trim segment.
- (14) DAMPER, RATI. This is the variable damper for TRIM. The purpose of this is to gradually dampen out oscillations of the trim iterations. This is accomplished by checking the errors generated in TRIM against an upper limit and, whenever all errors are less than this limit, reducing both the partial derivative increment and the maximum amount which one of the TRIM variables can change in one iteration. This second entry to this subroutine, RATI, limits and applies the corrections to the TRIM variables.
- (15) DAT1. This block data contains C_L , C_D , and C_M tables for the NACA 0012 airfoil.
- (16) DAT3. This block data subroutine contains the Carta tables used in subroutine CMCALC.
- (17) DERIV. This subroutine evaluates the derivatives during a maneuver.
- (18) DIFFER. This is a subroutine of numerical differential technique. Outputs from it are quantities of velocities and accelerations required for the calculation of unsteady aerodynamic effects.
- (19) ERRCHK. This subroutine checks possible input errors in the program logic group which is the key input group.
- (20) EXTORS. It recalculates cg location, inertias, and gross weight when any external store is dropped. It also updates aerodynamic brake locations if brake deployment during a maneuver is exercised.
- (21) FOCUS. This subroutine is in the rotor analysis. It transforms fuselage quantities into rotor reference and calculates accelerations during a maneuver.

- (22) FUSFNM. As the name implies, this subroutine computes fuselage aerodynamic forces and moments. It also calculates rotor nacelle drag contributions.
- (23) FUSINT. This subroutine converts input array to mnemonics for the fuselage group. It also calculates cg location and inertias if external stores are included.
- (24) GUST. This subroutine is entered only during a maneuver in which a gust is being generated. It calculates the distance of each part of the rotorcraft from the start of the gust and then calculates from that distance the magnitude of the gust velocity at each point on the ship.
- (25) HAMMS. This is the core of Hamming's Predictor-Corrector Method. It does predictions, checks errors, and makes corrections. It also handles the integration of differential equations and calls other subroutines necessary to a maneuver.
- (26) HARM, HARM1. The harmonic analysis for blade loads at the trim point is done by this subroutine.
- (27) HPCG. This subroutine calculates the first three time points. It is used along with trim points to start Hamming's Predictor-Corrector cycles in subroutine HAMMS.
- (28) HRESP. The elastic modes during the quasi-static trim procedure are handled by this subroutine.
- (29) INBLD, INBLD1, INBLD2, INBLD3. This subroutine converts input array to mnemonics for blade related data.
- (30) INIT. This subroutine initializes the arrays for the printout during a maneuver and also writes those arrays on disk for later plotting.
- (31) INRO. The function of this subroutine is the initialization and calculation of problem constants from the rotor inputs only.
- (32) INRTR. This subroutine initializes some of the rotor related data which are not done in subroutine INRO.
- (33) INSCAS. Initialization of the SCAS inputs is done here.
- (34) INSTAB. This subroutine initializes for a stability analysis. It calculates the partial derivatives needed for later computing the frequency response.
- (35) INTFRO. It interpolates blade natural frequency as a function of rotor rotational speed and blade collective pitch.

- (36) INVERS. This subroutine calculates the inverse of the mass matrix before the call to ALLMAT.
- (37) IOMAT. This subroutine prints the mass, damping, and stiffness matrices used in the stability analysis.
- (38) ITRIM. Included in this subroutine is the iteration loop of the trim section of the program. The function here is to iterate to a trimmed flight condition.
- (39) ITROT. This subroutine initializes variables for subroutine AZMUTH and, when specified by the input parameters, controls the iteration loops to balance the rotor flapping moments. It also calculates the trim conditions for an elastic blade.
- (40) JACOBI. As the name of this subroutine implies, its function is to calculate the Jacobian for use in the Newton-Raphson iteration method in TRIM or to calculate the displacement derivatives for use in the stability analysis.
- (41) JFBGIN, ATMINT. This subroutine converts input arrays to mnemonics for jets, flight constants, bobweight, and weapons groups.
- (42) JSTRED. This subroutine reads most of the input data groups.
- (43) LGCINT. The program logic group input array is converted to mnemonics in this subroutine.
- (44) LIZE. Initialization of some numerical constants is done in this subroutine.
- (45) LOADT. This subroutine computes and prints out the loads on an elastic blade at the trim point.
- (46) MAIN. This subprogram reads the control cards which direct the flow of the whole problem. The path is selected and calls are initiated to begin working the problem. Upon return, possible errors are checked for and appropriate action is taken. If an error is detected, an error message may be printed out. Then the program either terminates execution or starts the next problem, depending on the severity of the error.
- (47) MANU. This subroutine controls the time-variant maneuver segment if Runge-Kutta Method is used. It handles the integration of the differential equations and the calling of the other subroutines necessary to a maneuver.
- (48) MANUV. It controls the time-variant maneuver segment if Hamming's Method is used.

- (49) MATRIX. The function of this subroutine is to calculate the transformation matrix from a set of input Euler angles.
- (50) MBAL. This subroutine calculates rotor flapping increments during each iteration.
- (51) MDRDRS. Damping and stiffness matrices for a stability analysis are calculated here.
- (52) MNEM. The function of this subroutine is to calculate problem constants from input data and to initialize for a problem.
- (53) MODAL. The variables which are functions of mode shape, frequency, and mass and inertia distributions only are computed in this subroutine.
- (54) MODES. This subroutine calculates the mass matrix for a stability analysis.
- (55) MOMB, SUBA, SUBB. This subroutine simulates a servomechanism controlling the swashplate while the main rotor is being folded horizontally.
- (56) MTLT, TTLT, FLAT, YAWP. This subroutine handles mast tilt during a maneuver, the flat tracking mechanism, and the yaw control servomechanism.
- (57) NOPS, NOPS1. The inputs to this subroutine are the number of azimuth stations for use in the rotor analysis. The outputs are quantities which are functions of the number of azimuth stations.
- (58) NPUTOT. This subroutine prints most of the input data.
- (59) NSTED, NSTED1. This subroutine calculates some variables for unsteady aerodynamic effects.
- (60) PDZ1, PDZ. The inputs of this subroutine are a trim partial derivative matrix (i.e., the Jacobian) and the type of helicopter or rotorcraft being flown. This subroutine then changes the partial derivative matrix to conditions which are known to hold. Essentially, this is an attempt to filter numerical "noise" in the matrix.
- (61) PUNCH. It punches nonzero elements of mass, damping, and stiffness matrices used in the stability analysis. The form of the punched output cards is explained in Volume II.
- (62) PYLINT. It converts input arrays to mnemonics for the dynamic pylon group.
- (63) QUAN. This subroutine sets mnemonics from the integration array at the end of each time point.
- (64) RADBCN. It calculates some variables used by subroutine RADIAL.

- (65) RADIAL. This subroutine in the rotor analysis does the calculations and integration along the blade radius.
- (66) RADOUT. It prints output from subroutine RADIAL and AZMUTH.
- (67) RBDY. It initializes radial stations for a rigid blade.
- (68) READIN. This subroutine contains the logic for reading and printing the input data.
- (69) REDATB. It handles the read-in of airfoil data tables.
- (70) REDBMS. It handles the read-in of aeroelastic blade data as well as blade mode shape data.
- (71) REDCL. It reads the coefficients of lift, drag, and pitching moment of each airfoil data table.
- (72) REDID. It handles the read-in of group ID cards.
- (73) REDRWK, RWKOUT. It performs the read-in and printout of rotor-induced velocity distribution (RIVD) tables.
- (74) REDSWK, SWKOUT. It performs the read-in and printout of rotor wake at aerodynamic surface (RWAS) tables.
- (75) RESTR1, REST4, REST0, REST1, REST2. Restart tapes are written or copied by this subroutine. For the first case of each run, it also initializes all variables in commons to zero.
- (76) RGUST. Subroutine GUST carries wind gust calculations to the rotor hubs. This subroutine then carries the calculations to the blade elements.
- (77) RNGKTA. This is a special Runge-Kutta routine being used as a starter in Hamming's Predictor-Corrector Method.
- (78) ROTAN. This subroutine may be considered to be the outer section of the rotor analysis.
- (79) RPTPG. It controls the optional trim page output.
- (80) RTINIT. This is the control routine which handles the initialization of the rotor.
- (81) RTWAKE. This routine calculates blade local induced velocity when rotor wake table option is used.
- (82) SAVEIC, RESTOR, RESTR1. This routine saves initial conditions during calculation of the first three time points of Hamming's Method.

- (83) SAVOLD. It saves the values of some variables at the previous time point for Hamming's Method.
- (84) SCASIT. The highest derivatives in the differential equations for the SCAS are calculated here.
- (85) SIVAR. This subroutine handles the initialization of the maneuver inputs for subroutine VARI which are not a function of the trim point.
- (86) SOLVE. This subroutine solves systems of linear equations by Gaussian elimination.
- (87) STAB. This subroutine computes the rate derivatives used in the stability analysis.
- (88) START. The function of this subroutine is to change units of the input arrays and set them equal to mnemonics and to control the initialization segment.
- (89) STBWAK. This subroutine calculates the effect of rotor wakes on each stabilizing surfaces when a surface uses RWAS tables.
- (90) STBZFM. It calculates aerodynamic forces and moments at all stabilizing surfaces.
- (91) STBZIN. The function of this routine is the initialization and calculation of problem constants for wing and stabilizing surfaces.
- (92) SUPERP. This subroutine simulates an autopilot.
- (93) SWAS. This subroutine performs the function of linking the controls to the swashplates with the appropriate linkage factors and phase factors.
- (94) SWSRAT. It calculates some intermediate velocities and accelerations.
- (95) TABFIX. This subroutine calculates arrays to be used in the method of calculated entry in subroutine TABINT.
- (96) TABINT. As the name implies, this subroutine does a table interpolation for C_L , C_D , and C_M tables.
- (97) TABOUT. When tables are used for C_L , C_D , and C_M , this subroutine prints the tables after the input data.
- (98) TILT, HASF, TFFA. This subroutine controls cg shift calculations for several different manners of shifting cg. The primary function is in a mast tilt maneuver. It provides not only for cg shift but also for changes in control phasing as a function of the mast tilt angle. Secondary entries handle cg shift with folding of a rotor either when it is

being folded aft after being tilted forward and stopped or being folded horizontally after a stop.

(99) TIMLP. This subroutine contains the initialization necessary at the start of each time step.

(100) TIVAR. This subroutine handles the initialization of the maneuver inputs for subroutine VARI which are a function of the trim point.

(101) TRIM. As the name implies, this subroutine is the primary one of the section of the program for finding the trimmed flight condition.

(102) TRMANU. This routine sets up arrays for the outputs of trim as well as maneuver pages.

(103) TRM1. The primary function of this routine is to initialize variables used by time-variant trim.

(104) TURN. This subroutine handles a banked turn. Secondly, it handles pushovers or pull-ups. It does so by checking input data, picking up proper inputs, and doing the appropriate initialization to find a trimmed flight condition.

(105) TVT. This subroutine controls the time-variant trim procedure.

(106) UNSTED. This is the major section of the UNSAN unsteady aerodynamic model mentioned in Volumes I and II.

(107) VARI. This subroutine produces the effects of input disturbances during a time-variant maneuver. The inputs to this subroutine are the user-supplied forcing functions. The results produced by these functions are the output of this subroutine.

(108) VCNTRL, VCNT1, VCNT2, ---, VCNT10. This is called by subroutine VARI. It calculates the effects of time-variant maneuver disturbance.

(109) VIND. This subroutine calculates the induced velocity of a rotor.

(110) WAG. The time-dependent lift change by the Wagner and Kussner Method is computed in this subroutine.

(111) WING. As the name implies, this routine computes aerodynamic forces and moments on wings.

(112) WKTABN. If the number of blade radial stations input to rotor wake table is not twenty, this subroutine interpolates the table into twenty blade radial stations. This is done outside the iteration loops so that a three-way interpolation can be reduced to two-way.

- (113) WRFM. This is an output subroutine which writes out the rotor force and moment summary in shaft reference and the fuselage reference force and moment summary.
- (114) WRINST, IOWRFM, IOWRF1. This subroutine prints output during the computation of partial derivatives for a stability analysis.
- (115) WRMANU. This subroutine contains the write statements which produce part of the maneuver printout.
- (116) WRMDAL, WRMDL1, WRMDL2. This subroutine prints the results of subroutine MODAL.
- (117) WRMS. As a result of stability analysis, it prints out mode shapes associated with the rotorcraft characteristic roots.
- (118) WROT1, WROT. This is another output subroutine which produces the heading for the printout of the input data and the trim page.
- (119) WRSTAB. This subroutine prints the rate derivatives used in the stability analysis.
- (120) WRTMNV. This subroutine defines the output arrays for trim as well as maneuver pages.
- (121) WRTRIM. As the name implies, this routine writes trim page.
- (122) WRVP. This is still another output subroutine which produces the printouts of the partial derivative matrices calculated and the independent variables used in the calculation of those derivatives.
- (123) WSHDUF. It calculates fuselage effects on downwash and sidewash angles at wings and other stabilizing surfaces.
- (124) XCONIN. Initialization of all control linkages is handled by this subroutine.
- (125) XSTINT. This subroutine converts input array to mnemonics for external stores/aerodynamic brakes.
- (126) XSTORE. It calculates aerodynamic forces and moments at each external store/aerodynamic brake.
- (127) YRINIT. This subroutine conditions the aerodynamic inputs for the rotors.
- (128) YSINIT. This subroutine conditions the aerodynamic inputs for the wing and stabilizing surfaces.
- (129) ZERO. This is part of the initialization segment. Every variable in this routine is set to zero.

(130) ZLLCAL. This subroutine computes zero lift line increments at wings and other stabilizing surfaces.

3.3 ASSEMBLY LANGUAGE SUBPROGRAMS IN AGAJ73

AGAJ73 uses four assembly language subprograms which were written at Bell Helicopter Company: ABDUMP, DATE, DOTX, and PDSRED. The first three are available from the SHARE (an IBM user's organization) FORTRAN library. Although PDSRED itself is not available from SHARE, their library does include two routines (OPEN and CLOSE) which are generalizations of PDSRED and can be adapted to replace the single routine.

(1) ABDUMP. This diagnostic routine permits the program to terminate itself with a memory dump under OS/360. It does nothing except issue an ABEND macro with user code 0322. The routine is in System/360 Assembler Language. It was prepared at Bell Helicopter and is in the public domain.

(2) DATE. This routine with argument (NDATE) returns the current system date, NDATE, in Gregorian form: mm/dd/yy. NDATE must be at least eight bytes long. The routine is coded in System/360 Assembler Language for OS/360 only. It uses the macros peculiar to the MVT option of OS/360. It was prepared at Bell Helicopter and is in the public domain. It contains the following entry points:

ENTRY SETIME (TINT). This entry establishes an operating time interval against which to check program operation. This interval (TINT) is in minutes in floating point form. The routine does not cause execution to terminate at the end of the designated interval. This entry initializes TIMEX.

ENTRY TIMEX (TU, DT, TL). This entry checks the central processor time since the last call to SETIME or TIMEX. It returns three argument values in floating point minutes:

TU - Time since initial call to SETIME.

DT - Time since last call to TIMEX or SETIME.

TL - Time remaining in the SETIME interval.

(3) DOTX. This is a function subprogram with the following argument list: A, IA, B, IB, NP. It is a fast method of computing vector inner product using double precision accumulation of sum of products of single precision operands. The routine is physically small, about twice as fast as equivalent FORTRAN code, and quite flexible. It provides the improved accuracy of double precision vector products without the increase of storage and time required to do full double precision analysis. DOTX returns a double precision result if the calling program types it as such. The arguments are described as follows:

A and B are the first elements of the two single precision vectors to be multiplied.

IA is the number of words between successive elements of A. It is a full word integer greater than zero.

IB is the number of words between successive elements of B. It is a full word integer of any value.

NP is the number of element products to be formed, i.e., the number of elements used. It is a full word integer greater than zero.

The only validity check made on the arguments is that IA and NP be greater than zero. If either of them is out of range, the routine returns a zero result. There are no other error exits.

The computations of the subprogram can be represented by the following summation:

$$X = \sum_{i=0}^{NP} Y(NA + i*IA)*Z(NB + i*IB) \quad (3-1)$$

where

X is the output of the subprogram,

Y(NA) is the NAth element in the Y array (A in the calling sequence),

Z(NB) is the NBth element in the Z array (B in the calling sequence), and

NP, IA, and IB are as defined in the argument list.

For example, assume that the two arrays FORCE and DISP are dimensioned to 5 and 8 respectively. Using DOTX, the following statement could be programmed:

WORK = DOTX(FORCE(1),2,DISP(2),3,3) (3-2)

The equivalent FORTRAN statement would then be as follows:

WORK = FORCE(1)*DISP(2) + FORCE(3)*DISP(5) + FORCE(5)*DISP(8)

The routine is written in System /360 Assembler Language for any level of assembler. It was prepared at Bell Helicopter Company and is in the public domain.

(4) PDSRED. This routine is used to find a member of a partitioned dataset and to make it accessible to a FORTRAN routine through normal sequential READ statements. The parameter list is as follows:

member an 8-byte character string which identified the desired member

&sn the symbol & is a standard punch. The symbol sn stands for a statement number. If the member is not found, the program goes to this statement in the calling routine rather than executing the statement following the call to PDSRED. This type of argument is an IBM FORTRAN language extension.

After returning from PDSRED, normal FORTRAN READ on unit 1 may be issued and an end of dataset is sensed as with any other dataset. Before another call to PDSRED can be issued for the same PDS, Re-wind 1 must be issued.

This routine is coded in IBM System/360 Assembler Language using OS/360 macro facilities. This function is not duplicated in any other operating system. The routine is compatible with all OS/360 releases through and including the 21.6 release. It was prepared at Bell Helicopter and is in the public domain.

3.4 FORTAN SUBROUTINES IN GDAJ07

The FORTRAN subroutines contained in GDAJ07 are listed in alphabetical order including the main program, which is called MAIN. A few remarks are made for each subroutine which indicate its general use or purpose in the program.

(1) CALCS1. This subroutine is the interface between subroutine SCALIT and the CALCOMP plot routines.

(2) CONPLT. This subroutine controls the plotting of the time histories.

(3) CURVET. This subroutine analyzes the time history of selected variables during a maneuver. This analysis is accomplished by a least-squares curve fit followed by comparison of both the amplitude and phase angle of different variables. Then one variable is expressed as a linear function of two others.

(4) C81L. The function of this subroutine is the transfer to a disk of maneuver time history data which has been stored on a tape or the transfer to a tape of maneuver time history data which has been stored on a disk.

(5) DATP. This first block data subroutine contains part of the headings for plotted time histories.

(6) DAT1. This second block data subroutine contains part of the headings for plotted time histories.

- (7) FSFT. This subroutine controls the harmonic analysis of a time history.
- (8) HARM. This is the harmonic analysis subroutine used by subroutine FSFT.
- (9) HEDING. This subroutine generates the labels for the time histories from the data stores in DATP or DAT1.
- (10) MAIN. This is the control program for GADJ07.
- (11) PLOT. This subroutine does the CALCOMP plotting of the results of the harmonic analysis.
- (12) PLOT. This is the printer plot routine which produces plots of time histories.
- (13) SCALIT. This subroutine sets up the arrays for the time history plots.
- (14) SCLFIX. This subroutine calculates scale factors for the time history plots.
- (15) WROT1, WROT. This subroutine prints the headings on the printer plots.

3.5 LABELED COMMONS IN AGAJ73

There are 21 labeled COMMONS, but no blank COMMON, in AGAJ73. Each of the COMMONS is listed below. Any special ordering of variables and the reason for doing so are given along with some general comments.

- (1) ANDOIT. The first 9 variables, HPRC through YSHRN, in this COMMON are double precision. It is used only in the general purpose subroutines controlled by subroutine ANAL.
- (2) FLEX. It contains most of the variables used in elastic blade modal analysis.
- (3) FORCMC. This COMMON contains the Carta tables used by subroutine CMCALC. Since it is defined in block data format in subroutine DAT3, it is not marked as a COMMON block in Table 4-1.
- (4) FORWK. This COMMON contains most of the variables used in computing the rotor-induced velocity distribution from the table stored in FORWK1.
- (5) FORWK1. This is the set of rotor-induced velocity distribution (RIVD) tables used by subroutine RTWAKE.

- (6) FORY. There is no special ordering of variables in this COMMON. It consists of the array "Y" operated upon by Runge-Kutta and is used in the initialization, trim, and maneuver segments.
- (7) FORYYY. It contains most of the variables specifically used by Hamming's Predictor-Corrector Method.
- (8) FORYY1. This COMMON has only one array. It retains all the values of previous time points and their derivatives used by Hamming's method.
- (9) FOSWK. This COMMON contains most of the variables used in computing the effects of the rotor wake at the aerodynamic surfaces from the tables stored in FSWK1.
- (10) FOSWK1. This is the set of rotor wake at the aerodynamic surfaces (RWAS) tables used by subroutine STBWAK.
- (11) INSTAR. This block contains most of the input. It is used in the initialization segment.
- (12) MANAL. The first 59 variables, XF through NQTR, in this COMMON are ordered to allow I/O and other manipulations to be done on an equivalent array. The next 11 variables, COLSTK through B1TR, are ordered for equivalencing to an array. Not more than 10 of these variables are used, and the array KVAR is used as a pointer vector to choose which ones and their order. The next 22 variables, ALM through PYLMD, are ordered for equivalencing to the array VAR in subroutine STAB for the calculation of derivatives. The variables TAXL and TAXR are equivalenced to an array in subroutine CNTH.
- (13) STAMAN. Variables in this block are mostly used in initialization and maneuver segments. Arrays SHPCRP through SPTGRP are ordered to allow I/O and other manipulations to be done on an equivalent array.
- (14) STARAD. Most of the variables here are used in the initialization and general-purpose segments.
- (15) STARAN. This COMMON is used in the initialization and general-purpose segments.
- (16) STBD. This COMMON is used only in the stability analysis.
- (17) STRIAB. COMMON STRIAB is used in the initialization, trim, and stability analysis segments.
- (18) STRIMA. The first 16 variables, TZM through TCLOCK, are ordered for equivalencing in subroutine MOMB. The next six arrays, SCASRF through SCASYC, are ordered for equivalencing in subroutine INSCAS. This COMMON is not used in the rotor analysis.

(19) TAB. This COMMON contains the C_l table for the NACA 0012 airfoil. Since it is defined in block data format in Subroutine DAT1, it is not marked as a COMMON block in Table 4-1.

(20) TAB1. This COMMON contains C_l and C_d tables for the NACA 0012 airfoil. Like TAB, it is block data in DAT1. Consequently, it is not identified as a COMMON block in Table 4-1.

(21) TOPLOT. This COMMON is used in all segments.

3.6 LABELED COMMONS IN GDAJ07

There are 6 labeled COMMONS, but no blank COMMONS, in GDAJ07. Each of the COMMONS is listed below, together with pertinent comments.

(1) INPLOT. This COMMON is used by subroutine SCALIT and the other subroutines in the segment for plotting time histories.

(2) MAXMIN. It contains the maximum and minimum values of the specified variable. It is primarily used to determine the scale of the plot.

(3) PLOTD. This COMMON contains the data in the block data subroutine DATP which is used by subroutine HEDING to furnish alphameric headings for time histories.

(4) PLOTD1. It has the data in the second block data subroutine DAT1 which is used by subroutine HEDING to supply headings for time histories.

(5) TIMPTS. This COMMON is chiefly used as local storage in subroutines CURVET and C81L.

(6) TOPLOT. This COMMON contains control variables and is not the same as COMMON TOPLOT in AGAJ73.

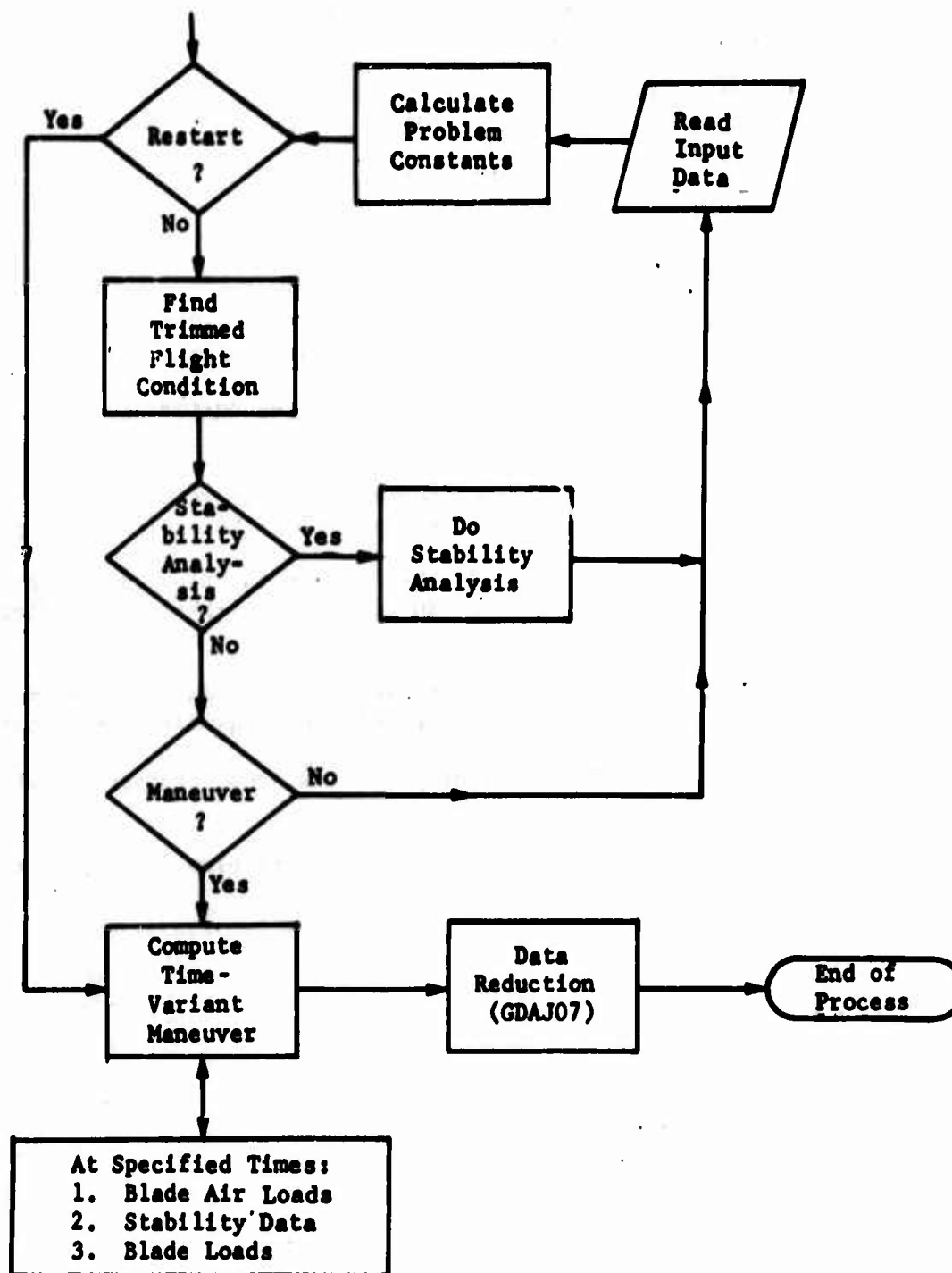


Figure 3-1. Flow Chart of System Structure.

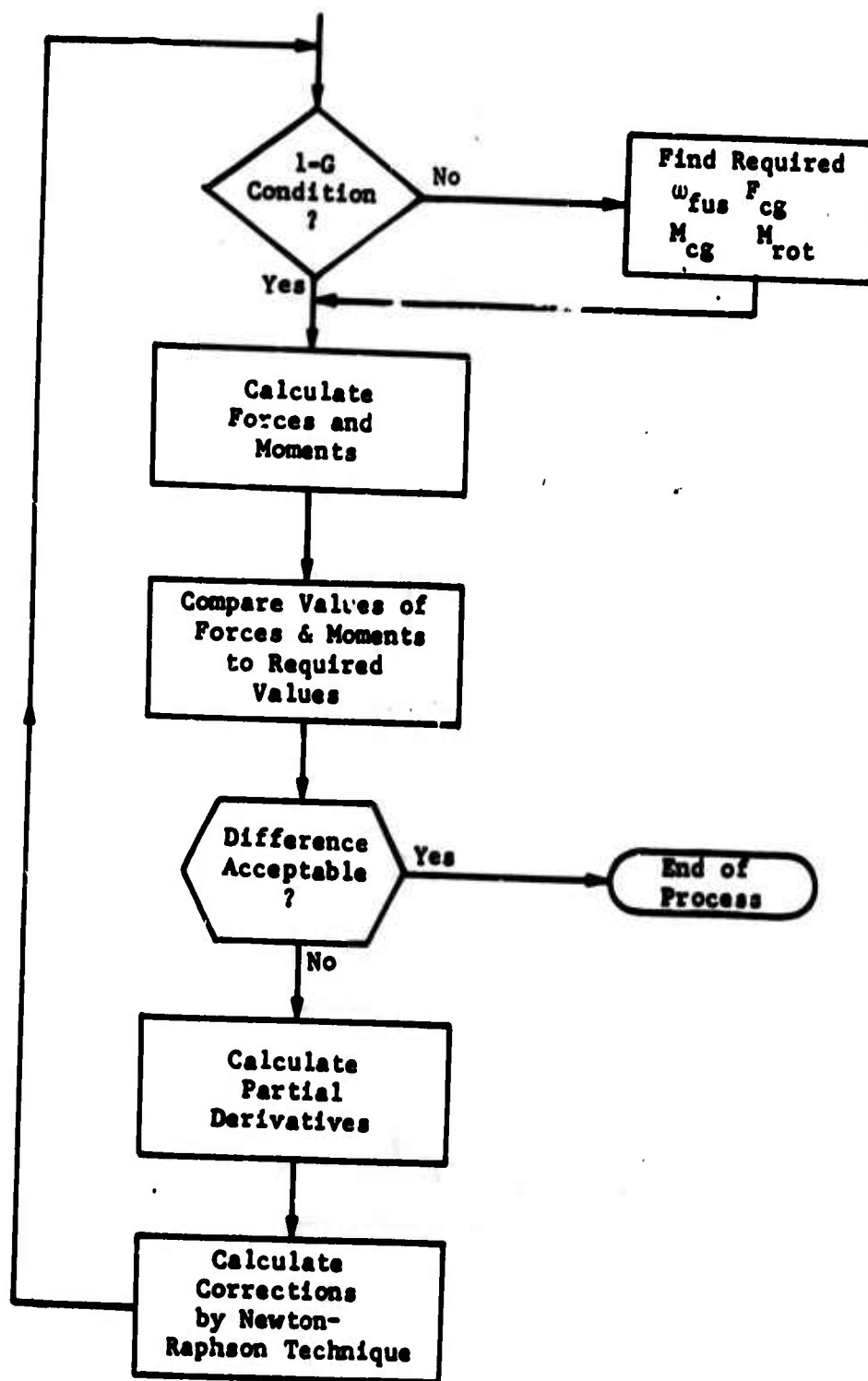


Figure 3-2. Flow Chart of Trim Process.

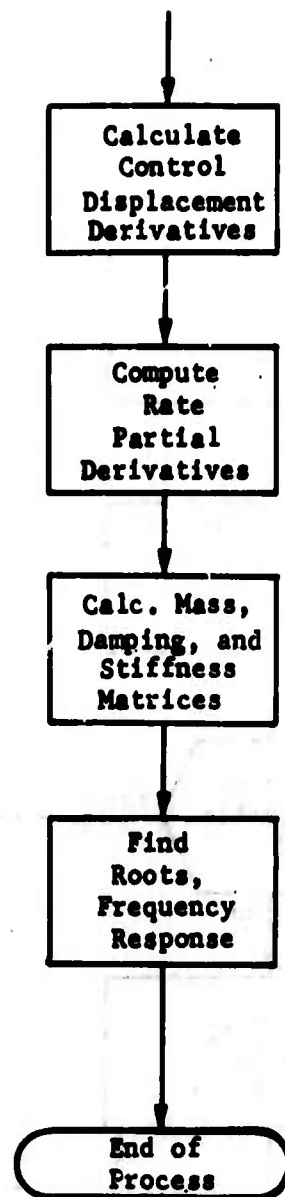


Figure 3-3. Flow Chart of Stability Analysis.

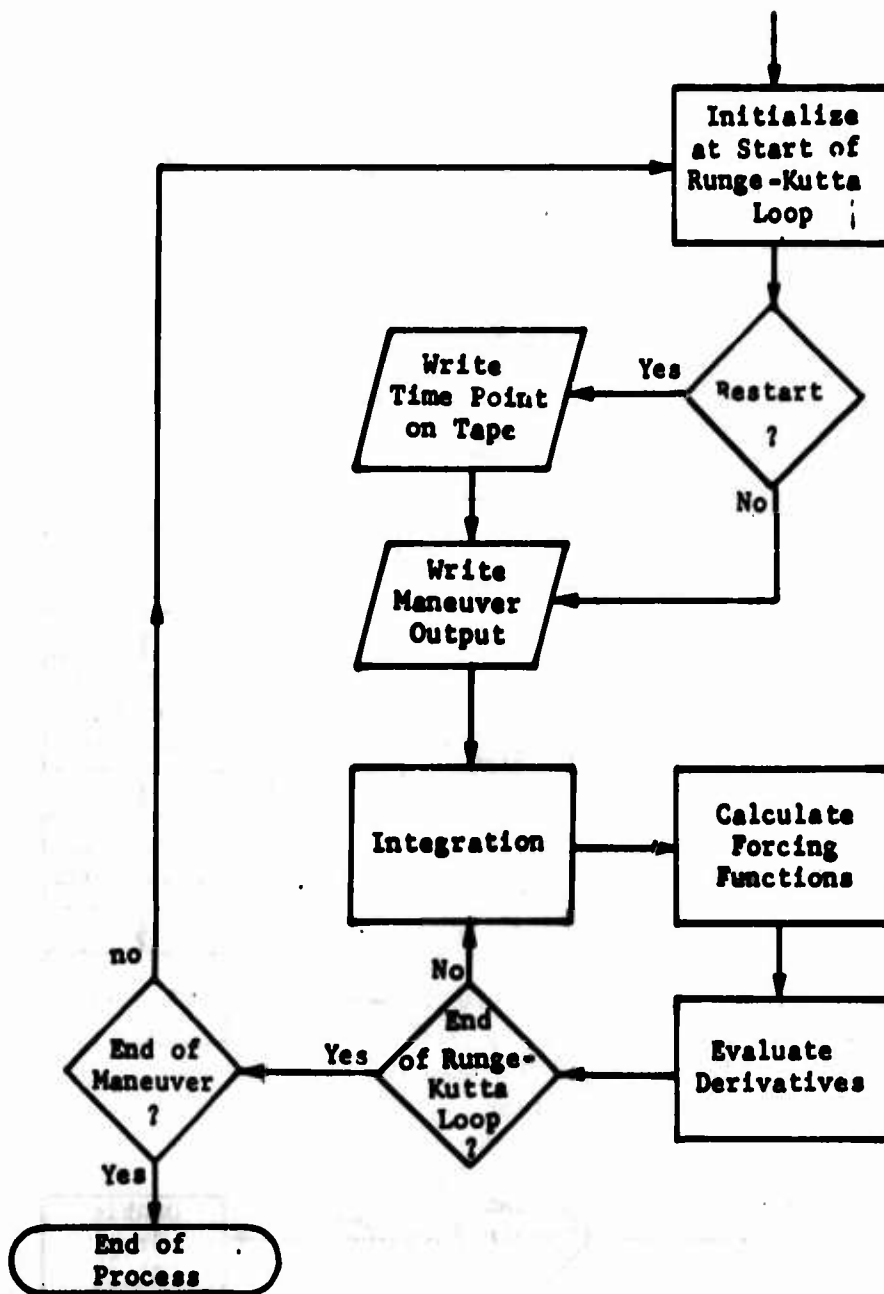


Figure 3-4. Flow Chart of Maneuver With Runge-Kutta Method.

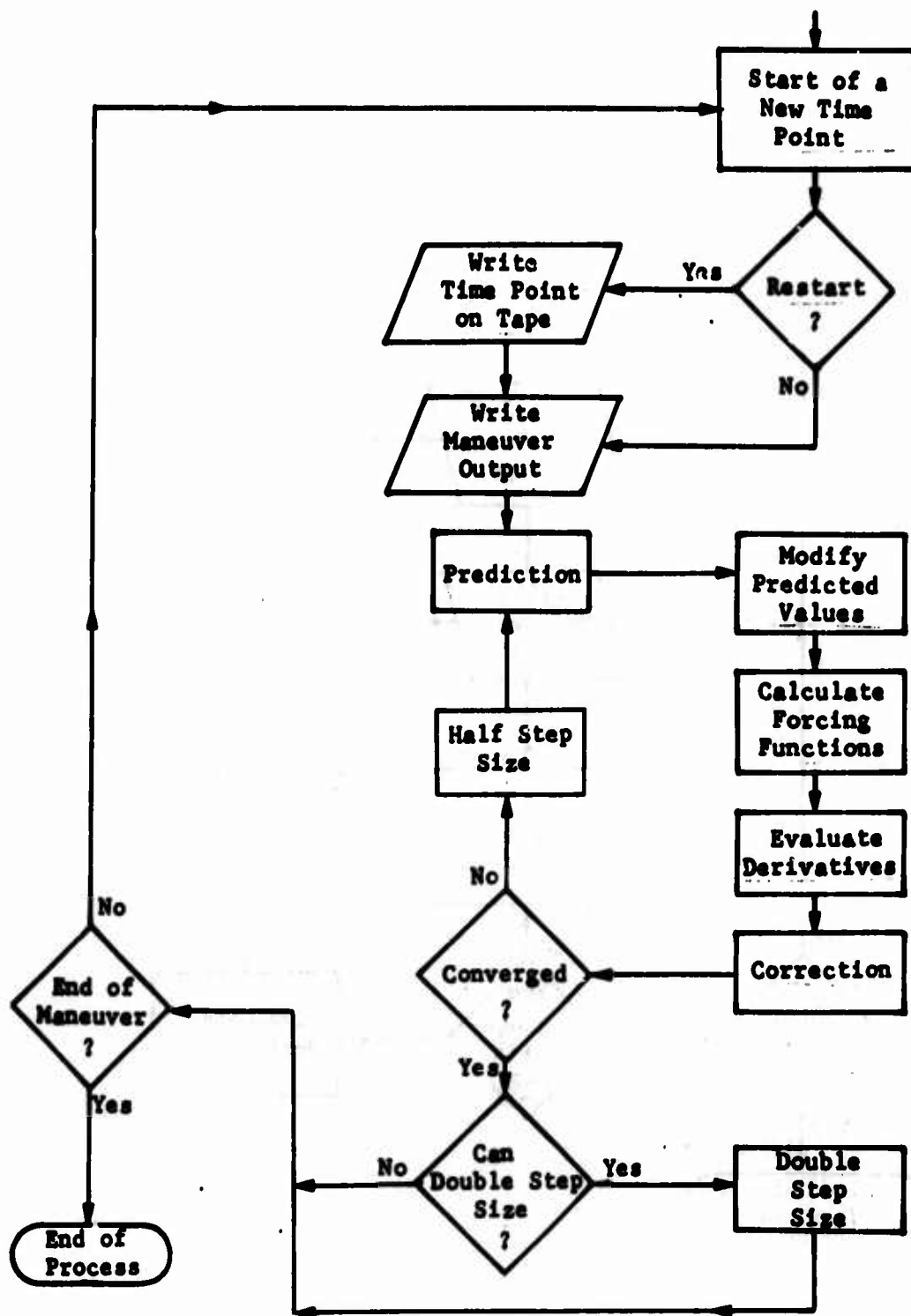


Figure 3-5. Flow Chart of Maneuver With Hamming's Method.

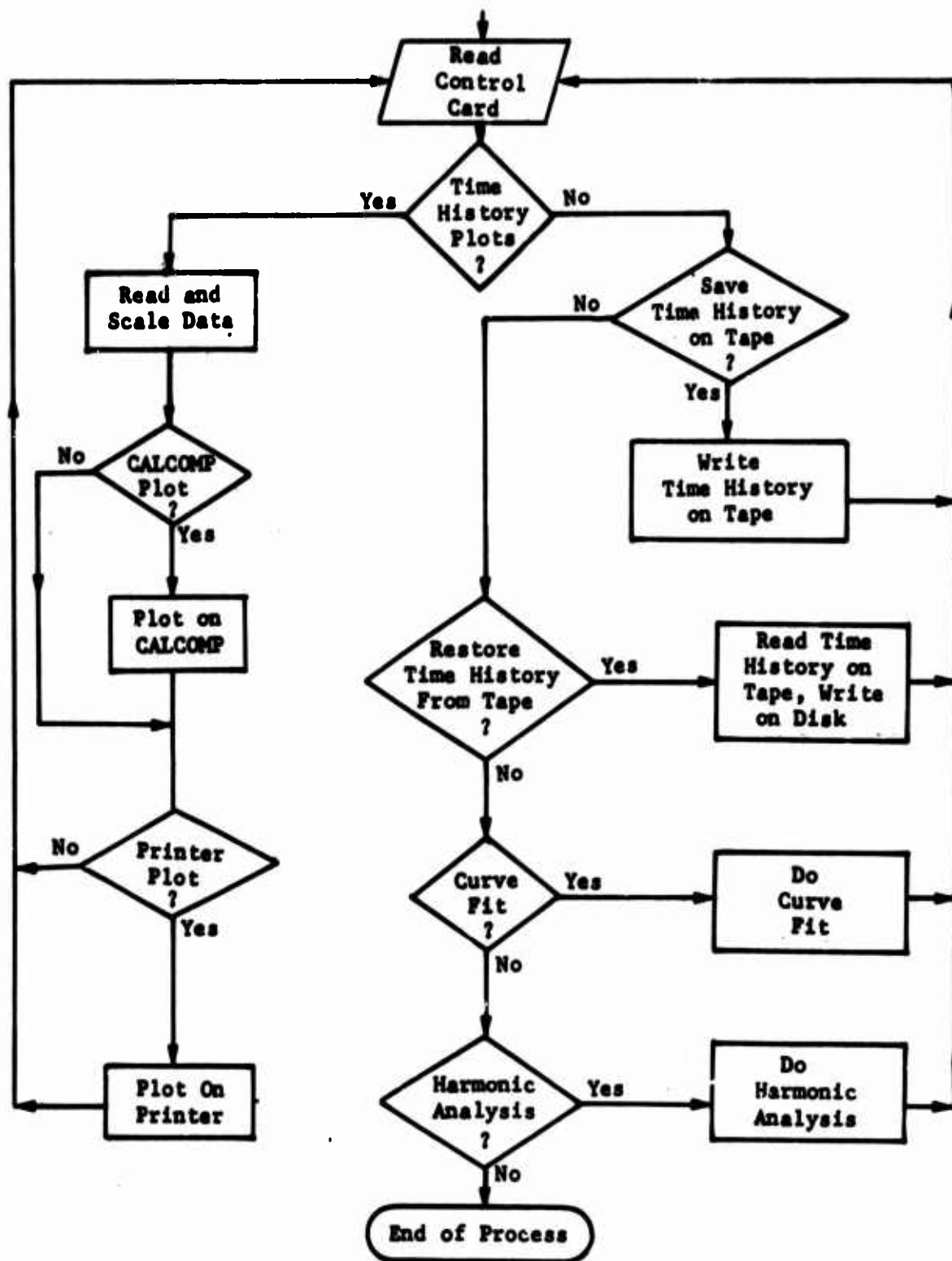


Figure 3-6. Flow Chart of GDAJ07.

4. DETAILED PROGRAMMING AIDS

4.1 CONTROL SECTION CROSS-REFERENCE

The Control Section Cross-Reference List for AGAJ73, Table 4-1, shows most of the control sections, including COMMONS, which are referenced by another control section. The exceptions are listed on the first page of the table. These exceptions are system routines, and their presence would not contribute to the cross-reference list. The remaining pages of Table 4-1 contain the control sections in alphabetical order in a column on the left side of the page. To the right of each control section name is the cross-reference information. "LENCT:" is the size of the subroutine or COMMON in hexadecimal bytes. "CALLED BY" gives the name of each control section referencing the control section whose name is in the column on the left. "IS USED BY" gives the name of each control section referencing a control section in the "CALLED BY" list or by another control section in the "IS USED BY" list. "CALLS" gives the name of each control section referenced by the control section whose name is in the column on the left. "USES" gives the name of each control section referenced by a control section in the "CALLS" list or by another control section in the "USES" list.

The information in the Control Section Cross-Reference List is sufficient to construct the sequence of subroutine calls from which an overlay structure can be made.

As noted in Section 3.2, several subroutines have multiple entry points. However, the Cross-Reference List (Table 4-1) includes only the primary names of subroutines; it does not include the names of any of these additional entry points. In the case where a call to a subroutine is actually a call to an additional entry point, the primary name of the subroutine which contains the specified entry point is used in the Cross-Reference List. For example, Table 4-1 indicates that subroutine AZMUTH calls RADOUT when AZMUTH actually calls AZMOUT (a second entry point to RADOUT). In Section 3.2 the names of any additional entry points follow the name of the subroutine; however, in reading the program listing, it is generally more useful to know the name of the subroutine which contains a particular entry point than to know the additional entry points of a subroutine. Hence, a cross-reference of the additional entry points to the subroutine which contains them is given in Table 4-2.

Table 4-3 contains the Control Section Cross-Reference List for GDAJ07. It is read and used in exactly the same manner as Table 4-1.

4.2 PROGRAM LISTINGS

Listings of the three programs supplied under this contract are on file at Eustis Directorate, USAAMRDL:

- (1) AGAJ73
- (2) GDAJ07
- (3) AS812A

Note that Program AS812A is written in PL/1 rather than FORTRAN.

Program AS812A is independent of the AGAJ73/GDAJ07 combination. The purpose of the program is to perform least-squared-errors curve fits of wind tunnel data which provide inputs for the Nominal Angle Equations of AGAJ73. The card format for the wind tunnel data input to AS812A was chosen to be compatible with the format of data supplied by the LTV Low Speed Wind Tunnel, Dallas, Texas. The data is read in by the statement on CARD 500 for the sequenced source deck. The format may be changed by the programmer to read most any available data.

The printed output of AS812A lists the coefficients of the equation to which the data has been fitted. The output also lists the inputs to AGAJ73 in the proper sequence and format with the AGAJ73 card number. Punched card output, for direct input to AGAJ73, is an option in AS812A. The job control cards for AS812A are shown in Figure 4-1.

4.3 PROGRAMMING OPTION IN NUMERICAL INTEGRATION

As mentioned in Section 2 of Volume I, there are two methods of numerical integration techniques in AGAJ73. The first one is the Runge-Kutta method which is programmed as the default option. The second is Hamming's method which can be activated only by setting the logical variable HMG to TRUE. This variable is located in subroutine LIZE and is set to FALSE in the program delivered under this contract.

4.4 REDUCTION OF PROGRAM MEMORY REQUIREMENTS

There are several very large arrays in AGAJ73. The size of some can be easily reduced without any significant decrease in the capabilities of the program. The arrays which can be easily reduced in size are shown in Table 4-4. The table also includes the name of the COMMON to which each belongs, the full and reduced dimensions of each, and the approximate savings in memory requirements for reducing each array. Up to 330K of storage can be eliminated with minimal effort.

Referring to Table 4-3; arrays WKTR, WKSTB, and AUX are independent of each other. Their sizes can be cut individually or concurrently. However, arrays CURVED, CURVEL, and CURVEM are dependent; if one of them is reduced, all of them must be reduced at the same time. If, after the program is link-edited in load-module form, the programmer wishes to reduce the size of any array noted in Table 4-4, it is necessary to include a REPLACE card, before the INCLUDE OLDLIB or SYSLMOD card in the linkage step. This REPLACE card must then contain the names of all arrays which have just been reduced in size.

In Table 4-4 there are two columns under the headings of array name and dimension. The "Outride RESTRT" column gives the name(s) and dimension(s) to be used in the associated COMMON block when it is included in any subroutine other than RESTRT. In subroutine RESTRT the information listed under "In RESTRT" must be used. The different names and single dimensions are in RESTRT simply to facilitate the task of setting the COMMON blocks to zero. Each array included in Table 4-4 is discussed in more detail below.

WKRTTR contains the sets of coefficients for the Rotor-Induced Velocity Distribution (RIVD) tables discussed in Sections 2.3.3.1.3 and 2.3.3.2 of Volume II. If the size of this array is reduced as shown in Table 4-4, the input to IPL(5) must be equal to zero and in subroutine RESTRT FRK1(66000) and DO 34 I=1,66000 must be changed to FRK1(1) and DO 34 I=1,1, respectively.

WKSTB contains the sets of coefficients for the Rotor Wake at Aerodynamic Surface tables discussed in Section 2.3.4 of Volume II. If the size of this array is reduced as shown in Table 4-4, the input to IPL(37) must be equal to zero and the inputs to XWG(29) through XWG(32), XSTB1(29), and XSTB1(32) must all be less than or equal to 100. XSTB1 stands for the 1th stabilizing surface. In subroutine RESTRT, FSK1(9000) and DO 36 I=1,9000 must be changed to FSK1(1) and DO 36 I=1,1, respectively.

CURVED is the array where drag coefficients of the airfoil data tables are located. If the size of this array is reduced as shown in Table 4-4, the input to IPL(2) must be less than or equal to 2, TABL(6350) in subroutine RESTRT must be changed to TABL(3026), and TITLE(8,4) and CURVED(1100,4) in block data DAT1 must be changed to TITLE(8,1) and CURVED(1100,1), respectively.

CURVEL and CURVEM are the arrays for the airfoil lift and pitching moment coefficients respectively. If their sizes are reduced as shown in Table 4-4, the input to IPL(2) must be less than or equal to 2, TABL1(5375) in subroutine RESTRT must be changed to TABL1(2150), and CURVEL(1100,4) and CURVEM(575,4) in block data DAT1 must be changed to CURVEL(1100,1) and CURVEM(575,1), respectively.

AUX is an auxiliary array used in the Hamming's method. If this array is reduced as shown in Table 4-4, the default option of numerical integration in maneuver (the Runge-Kutta method) must be used. Also, in subroutine RESTRT, FYY1(3440) and DO 104 I=1,3440 must be changed to FYY1(1) and DO 104 I=1,1, respectively.

There are other large arrays which can be reduced if storage requirements are still excessive after taking the steps discussed above. However, the necessary steps for reducing them are more complicated and in some cases mean deletion of a major program option, and in most cases the savings is not worth the effort. The programmer should consult with the contractor before attempting any storage reductions beyond those outlined in this report.

4.5 SWITCH FOR DIAGNOSTIC DATA FROM STAB

In Section 3.2 of Volume II, IPL(34) is defined as a switch for obtaining diagnostic data during the stability analysis (STAB). Since the data generated by this switch are not of general interest to the user, but can be useful to the programmer, the function of IPL(34) is discussed in this Programmer's Manual rather than in Volume II. The function of the switch is described below.

In STAB there are up to 22 independent variables which may be incremented in the process of computing the stability (partial) derivatives. The number of variables actually incremented depends on the number of degrees of freedom which the user has activated. (See IPL(30) and (32) in Section 3.2 of Volume II). In each STAB case, IPL(34) can be used to print out the following data resulting from one of the variables being incremented:

- (1) blade element aerodynamic data (α , C_L , C_D , C_M , etc.)
at each blade station and each azimuth location for each rotor (i.e., IPRINT in subroutine RADIAL does not equal zero, which calls RADOUT)
- (2) rotor and pylon moment data (i.e., CONDI in subroutine MBAL is greater than 1.5, which causes printout)

To generate this output for a particular increment, IPL(34) is set to the value shown in Table 4-5. Further information about the variables in this table can be found in Section 4.11.2.1 of Volume II. Note that locking out a degree of freedom does not change the correspondence shown in Table 4-5 between IPL(34) and the variables. Also, it is only possible to obtain this extra printout for one variable in each STAB case. To obtain the printout for more than one variable, the case must be rerun for each variable of interest with IPL(34) set to the appropriate value in each repeat run.

TABLE 4-1. CONTROL SECTION CROSS-REFERENCE FOR AGAJ73

PURG INCLCLABS	CONTROL SECTION	REMOVED
PURG INCLCLAS	CONTROL SECTION	REMOVED
PURG INCLCLSQT	CONTROL SECTION	REMOVED
PURG INCCOMH2	CONTROL SECTION	REMOVED
PURG INCCSABS	CONTROL SECTION	REMOVED
PURG INCCSAS	CONTROL SECTION	REMOVED
PURG INCECOMH	CONTROL SECTION	REMOVED
PURG INCEFIOS	CONTROL SECTION	REMOVED
PURG INCEFNTH	CONTROL SECTION	REMOVED
PURG INCERRM	CONTROL SECTION	REMOVED
PURG INCETRCH	CONTROL SECTION	REMOVED
PURG INCFCVTH	CONTROL SECTION	REMOVED
PURG INCFOPT	CONTROL SECTION	REMOVED
PURG INCFRXPR	CONTROL SECTION	REMOVED
PURG INCLSORT	CONTROL SECTION	REMOVED
PURG INCNAMEL	CONTROL SECTION	REMOVED
PURG INCSASCN	CONTROL SECTION	REMOVED
PURG INCSATN2	CONTROL SECTION	REMOVED
PURG INCSEXP	CONTROL SECTION	REMOVED
PURG INCSLOG	CONTROL SECTION	REMOVED
PURG INCSSEN	CONTROL SECTION	REMOVED
PURG INCSSORT	CONTROL SECTION	REMOVED
PURG INCSTNCT	CONTROL SECTION	REMOVED
PURG INCUATBL	CONTROL SECTION	REMOVED
PURG INCUOPT	CONTROL SECTION	REMOVED

GROUP	LENGTH CALLED BY	20 - MAIN
AJACOB	LENGTH 878 CALLED BY - INSTAB IS USED BY - CONSTB CALLS - ANAL USES - ANDOIT	JACOB1 INSTAB FORVY1 ZLLCAL AZMUTH FORV1 FORV1 INSTED STRINA W5H0UF WING
ALLMAT	LENGTH 4588 CALLED BY - ALSTAB IS USED BY - CONSTB	MAIN
ALSTAB	LENGTH 6420 CALLED BY - CONSTB IS USED BY - MAIN CALLS - ALLMAT USES - STARAN	IMVERS STBO
ANAL	LENGTH 920 CALLED BY - AJACOB IS USED BY - CONSTB CALLS - ANDOIT USES - ANDOIT	MAIN STBO IMVERS STBO
ANDOIT	LENGTH 400 CALLED BY - ANAL IS USED BY - AJACOB CALLS - ANDOIT USES - ANDOIT	THIS IS A "COMMON" CONTROL SECTION ANAL AZMUTH RADIOG WING
AZMUTH	LENGTH 740 CALLED BY - AZMUTH IS USED BY - AJACOB CALLS - ANDOIT USES - ANDOIT	ANAL MAIN FLEX
AZMUTH	LENGTH 460 CALLED BY - ITR0T IS USED BY - AJACOB CALLS - ANDOIT USES - ANDOIT	ITR0T JACOB1 MAIN FLEX

TABLE 4-1. Continued.

CDCL	LENGTH 898 CALLED BY - RADIAL IS USED BY - AJACOB CALLS - ANDOIT USES - ANDOIT	UNSTED ANAL MAIN MANAL TAB	AZMUTH MANU STARAD TAB1	CONSTB MANU STARAN	CONTRM RADIAL TABINT	DERIV RNGKTA	FOCUS ROTAN	HAMMS STAB	HPCG TRIM	INSTAB TVT	ITRIM	ITROT
CGZRM	LENGTH 700 CALLED BY - EXTORS IS USED BY - DERIV CALLS - INSTAR	TILT HAMMS MANAL	HPCG STAMAN	MAIN STRIMA	MANU TOPLOT	MANUV	MNEM	MOMB	NTLT	RNGKTA	START	VARI
CLCD	LENGTH EFB CALLED BY - STRZFM IS USED BY - AJACOB CALLS - ANDOIT USES - ANDOIT	WING ANAL RNGKTA MANAL TAB	CONSTB STAB STARAN TAB1	CONTRM TRIM STRIB	DERIV TABINT	HAMMS TOPLOT	HPCG	INSTAB	ITRIM	JACOBI	MAIN	MANU
CMCALC	LENGTH 600 CALLED BY - RADIAL IS USED BY - AJACOB CALLS - JACOBI USES - ANDOIT	ANAL MAIN FORCHC	AZMUTH MANU	CONSTB MANUV	CONTRM RNGKTA	DERIV ROTAN	FOCUS STAB	HAMMS TRIM	HPCG TVT	INSTAB	ITRIM	ITROT
CNTM	LENGTH 610 CALLED BY - VARI IS USED BY - DERIV CALLS - MANAL	HAMMS STAMAN	HPCG STRIMA	MAIN TOPLOT	MANU	MANUV	RNGKTA					
CONSTB	LENGTH 260 CALLED BY - MAIN CALLS - ALSTAB USES - AJACOB DOTX HPESP PUNCH STBZFM WAG	INSTAB ALLMAT FLEX INSTAR RADBN STRIB WING	MANAL ANAL FOCUS INTERQ RADIAL STRIMA WRFM	MODES ANDOIT FORCHC INVERS RADOUT SWAS WRINST	STAB AZMINT FORWK IONAT RGUST SHSRAT WRMS	STARAN AZMUTH FORWK ITROT ROTAN TAB WRSTAB	STBD CDCL FORV JACOBI RTMAKE TABINT WRVP	STRIMA CLCD FORVYI MANAL STAMAN TAB1 WSPDOUF	TOPLOT CMCALC FORVYI MATRIX STARAD TOPLOT XSTORE	DAMPER FOSWK MEAL STARAN TVT ZLLCAL	DATE FOSWK1 MDRDS STBD UNSTED	DIFFER FUSFNM NSTED STBWK VIND
CONTRM	LENGTH 110 CALLED BY - MAIN CALLS - LADT USES - AJACOB FLEX HRESP RADBN STBZFM TVT ZLLCAL	STRIB ANAL FOCUS INSTAR RADIAL STRIB UNSTED	TOPLOT ANDOIT FORCHC INTERQ RADOUT STRIMA VIND	TRIM AZMINT FORWK ITRIM RGUST SWAS WAG	AZMUTH FORWK ITROT ROTAN SHSRAT WING	CDCL FORV JACOBI RTAKE WRFM	CLCD FORVYI MANAL RTMAKE TABINT WRGT1	CMCALC FORVYI MATRIX SOLVE TAB1 WRTHNV	DAMPER FOSWK MRAL STAMAN TIVAR WTRIM	DATE FOSWK1 NOPS STARAD TOPLOT WRVP	DIFFER FUSFNM NSTED STARAN TRMANU WSPDOUF	DOTX FOSWK NSTED STBWK TRMI XSTORE
DAMPER	LENGTH 368 CALLED BY - INSTAB IS USED BY - CONSTB CALLS - STARAN,	ITRIM CONTRM STRIB	TRIM MAIN STRIMA									

DATE	LENGTH CALLED BY IS USED BY	1FO - ITRIM - CONSTB TIMLP	MAIN CONTRM TRIM	PUNCH MAMU MTRIM	URAMU MPCG	WROT1 INIT	WTRIM MAIN	ROMU	MAMU	ROMES	READIN	RPTG	START
DERIV	LENGTH 090 CALLED BY - MPMG IS USED BY - MPMG CALLS - ANAL USES - ANCOIT FOCUS INSTAR INTFRO RADOUT SUPERB WAG		MPCG FORV AZINT FORMC INTFRO JACOB SHAS WING	MAMU MAMU FORVY AZINT FORMC INTFRO JACOB SHAS WING	RMKTA FORVY COCL FORVY INTFRO JACOB SHAS WING	ANAL COZAR FORVY MAMU SOLVE TABINT ZLLCAL	QUAN CLOD FORVY MAMU STAB TAB1	ROTAN CHCALC FORVY MAMU STAB TAB1	SCASIT COTM FORVY MAMU STAB TAB1	STANAM DIFFER FORVY MAMU STAB TAB1	STRINA DOTX FORVY MAMU STAB TAB1	TOPLOT EXTORS GUST RADON STRINA VINTL	VARI PLEX WRESP RADIAL STRINA VINTL
DIPPER	LENGTH 3AP CALLED BY - MPMG IS USED BY - MPMG CALLS - ANAL USES - ANCOIT FOCUS INSTAR INTFRO RADOUT SUPERB WAG		MPCG FORV AZINT FORMC INTFRO JACOB SHAS WING	MAMU MAMU FORVY AZINT FORMC INTFRO JACOB SHAS WING	RMKTA FORVY COCL FORVY INTFRO JACOB SHAS WING	ANAL COZAR FORVY MAMU SOLVE TABINT ZLLCAL	QUAN CLOD FORVY MAMU STAB TAB1	ROTAN CHCALC FORVY MAMU STAB TAB1	SCASIT COTM FORVY MAMU STAB TAB1	STANAM DIFFER FORVY MAMU STAB TAB1	STRINA DOTX FORVY MAMU STAB TAB1	TOPLOT EXTORS GUST RADON STRINA VINTL	VARI PLEX WRESP RADIAL STRINA VINTL
DOTX	LENGTH 50 CALLED BY - FOCUS IS USED BY - FOCUS CALLS - JACOB WING		ANAL MAMU	ANAL MAMU	CONSTR MAMU	CONTRM RADIAL	DERIV RMKTA	FOCUS ROTAN	MAMU STAB	MPCG TRIS	INSTAB TVT	ITRIM	ITROT TVT
EMCME	LENGTH 490 CALLED BY - MPMG IS USED BY - MPMG CALLS - ANAL USES - ANCOIT FOCUS INSTAR INTFRO RADOUT SUPERB WAG		MPCG FORV AZINT FORMC INTFRO JACOB SHAS WING	MAMU MAMU FORVY AZINT FORMC INTFRO JACOB SHAS WING	RMKTA FORVY COCL FORVY INTFRO JACOB SHAS WING	ANAL COZAR FORVY MAMU SOLVE TABINT ZLLCAL	QUAN CLOD FORVY MAMU STAB TAB1	ROTAN CHCALC FORVY MAMU STAB TAB1	SCASIT COTM FORVY MAMU STAB TAB1	STANAM DIFFER FORVY MAMU STAB TAB1	STRINA DOTX FORVY MAMU STAB TAB1	TOPLOT EXTORS GUST RADON STRINA VINTL	VARI PLEX WRESP RADIAL STRINA VINTL
*FIBRES	LENGTH 900 CALLED BY - VARI IS USED BY - VARI CALLS - JACOB WING		MPCG FORV AZINT FORMC INTFRO JACOB SHAS WING	MAMU MAMU FORVY AZINT FORMC INTFRO JACOB SHAS WING	RMKTA FORVY COCL FORVY INTFRO JACOB SHAS WING	ANAL COZAR FORVY MAMU SOLVE TABINT ZLLCAL	QUAN CLOD FORVY MAMU STAB TAB1	ROTAN CHCALC FORVY MAMU STAB TAB1	SCASIT COTM FORVY MAMU STAB TAB1	STANAM DIFFER FORVY MAMU STAB TAB1	STRINA DOTX FORVY MAMU STAB TAB1	TOPLOT EXTORS GUST RADON STRINA VINTL	VARI PLEX WRESP RADIAL STRINA VINTL
FLEX	LENGTH 1850 CALLED BY - MPMG IS USED BY - MPMG CALLS - ANAL USES - ANCOIT FOCUS INSTAR INTFRO RADOUT SUPERB WAG		MPCG FORV AZINT FORMC INTFRO JACOB SHAS WING	MAMU MAMU FORVY AZINT FORMC INTFRO JACOB SHAS WING	RMKTA FORVY COCL FORVY INTFRO JACOB SHAS WING	ANAL COZAR FORVY MAMU SOLVE TABINT ZLLCAL	QUAN CLOD FORVY MAMU STAB TAB1	ROTAN CHCALC FORVY MAMU STAB TAB1	SCASIT COTM FORVY MAMU STAB TAB1	STANAM DIFFER FORVY MAMU STAB TAB1	STRINA DOTX FORVY MAMU STAB TAB1	TOPLOT EXTORS GUST RADON STRINA VINTL	VARI PLEX WRESP RADIAL STRINA VINTL
FOCUS	LENGTH 808 CALLED BY - ROTAN IS USED BY - ROTAN CALLS - JACOB WING		MPCG FORV AZINT FORMC INTFRO JACOB SHAS WING	MAMU MAMU FORVY AZINT FORMC INTFRO JACOB SHAS WING	RMKTA FORVY COCL FORVY INTFRO JACOB SHAS WING	ANAL COZAR FORVY MAMU SOLVE TABINT ZLLCAL	QUAN CLOD FORVY MAMU STAB TAB1	ROTAN CHCALC FORVY MAMU STAB TAB1	SCASIT COTM FORVY MAMU STAB TAB1	STANAM DIFFER FORVY MAMU STAB TAB1	STRINA DOTX FORVY MAMU STAB TAB1	TOPLOT EXTORS GUST RADON STRINA VINTL	VARI PLEX WRESP RADIAL STRINA VINTL

[illegible]

[illegible]

[illegible]

TABLE 4-1. Continued.

ITRIN	LENGTH 708 CALLED BY - TRIN IS USED BY - CONTR CALLS - AJACOB WRVP USES - AJACOB FORCPC ITROT STARAD TVT	MAIN DAMPER	DATE	JACOBI	MANAL	POBZ	SRLVE	S: MAN	STARAN	STRIB	STRINA	TOPLOT
		ANAL FORWK MANAL STARAN UNSTED	ANDOIT FORWK1 MATRIX STBWK VIND	AZINT FORV MBAL STRZFM WAG	AZMTH FORVYV NSTED STRIB WING	COCL FORVYV RADON STRINA WRFP	CLCD POSKE RADIAL SNAS WRVP	CHEALC FUSPM RADON SUSLAT WSTOR	DIFFER PUSPM RGUST TAB WSTOR	DOTX MRESP ROTAN TABINT ZLLCAL	FLEX INSTAR RTAKE TAB1	FOCUS INSTAR STARAD TOPLOT
ITROT	LENGTH 1348 CALLED BY - FOCUS IS USED BY - AJACOB NAMUV CALLS - ANDOIT USES - ANDOIT NSTED TAB1	ANAL RNGTA AZMTH AZINT RADON TOPLOT	CONSTB ROTAN FORWK COCL RADIAL UNSTED	CONTH STAB FORWK1 CHEALC RADOUT	DERTV TRIN MRESP DIFFER RGUST	MANMS TVT MANAL DOTX RTAKE	MPCE MBAL FLEX STARAN	INSTAB	ITRIN	JACOBI	MAIN	NAME
JACOBI	LENGTH 348 CALLED BY - INSTAB IS USED BY - CONSTB CALLS - AJACOB USES - ANAL FORWK MANAL STARAN UNSTED	ITRIN CONTH MANAL ANDOIT FORWK1 MATRIX STBWK VIND	MAIN STARAN AZMTH FORV MBAL STRZFM WAG	TRIN STRIB AZMTH FORVYV NSTED WING	STRINA COCL FORVYV RADON STRINA WRFP	TOPLOT CLCD FOSKE RADIAL SNAS WRVP	CHEALC FOSKE RADOUT SUSLAT WSTOR	DIFFER FUSPM RGUST TAB WSTOR	DOTX MRESP ROTAN TABINT ZLLCAL	FLEX INSTAR RTAKE TAB1	FOCUS INSTAR STARAD TOPLOT	FORCPC ITROT STARAD TVT
JFBCIN	LENGTH 980 CALLED BY - START IS USED BY - MAIN CALLS - INSCAS USES - STARAN	INSTAR STRINA	MANAL	MATRIX	STARAN	STARAD	STARAN	STRIB	STRINA	TOPLOT		
JOB	LENGTH 48 CALLED BY - MAIN											
JSTRED	LENGTH D78 CALLED BY - READIN IS USED BY - MAIN CALLS - INSTAR USES - FORWK	START MANAL FORWK1	RDY FOSKE	REDATB FOSKE1	REDOMS INSTAR	REDIO PORED	REDOMK REDCL	REDOMK REDIO	STARAN TAB	STARAD TAB1	STRINA TOPLOT	TOPLOT INSTAR
LGCTNT	LENGTH 450 CALLED BY - START IS USED BY - MAIN CALLS - INSTAR	MANAL	STARAD	STARAN	STRIB	TOPLOT						
LIZE	LENGTH 968 CALLED BY - STARY IS USED BY - MAIN CALLS - FLEX USES - FLEX	FORWK STRINA FORWK	FORWK1 TOPLOT FORWK1	FORV ZERO MANAL	FORVYV STARAN	FORVYV STRIB	POSKE STRINA	POSKE1 FOSKE1	INSTAR	MANAL	STARAN	STARAN

TABLE 4-1. Continued.

LENGTH 9630
 CALLED BY - CONTR
 IS USED BY - MAIN
 CALLS - FLEX

LENGTH 4F8
 CALLS - AROUND
 USES - AJACOB
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

DATE
 DIFFER
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

MANAL
 CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STAMAN
 DATE
 DIFFER
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STARAN
 JOB
 ADJOIT
 DOTS
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STRIAB
 MANU
 AZHINT
 ERMCK
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

TOPLOT
 CMCALC
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

WROTI
 CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

LENGTH 558
 CALLED BY - MAIN
 CALLS - DERIV
 USES - ANAL
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

DATE
 DIFFER
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

MANAL
 CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STAMAN
 DATE
 DIFFER
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STARAN
 JOB
 ADJOIT
 DOTS
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STRIAB
 MANU
 AZHINT
 ERMCK
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

TOPLOT
 CMCALC
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

WROTI
 CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

LENGTH 128
 CALLED BY - MAIN
 CALLS - MPCC
 USES - ANAL
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

DATE
 DIFFER
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

MANAL
 CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STAMAN
 DATE
 DIFFER
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STARAN
 JOB
 ADJOIT
 DOTS
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

STRIAB
 MANU
 AZHINT
 ERMCK
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

TOPLOT
 CMCALC
 FOCUS
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

WROTI
 CONTRA
 ALSTAB
 DERIV
 FURY
 INKLD
 JACOB
 MODS
 RADIAL
 RINGTA
 STARAD
 STARAD
 TABFIZ
 TVT
 WMS
 ZERO

TABLE 4-1. Continued.

MATRIX	LENGTH CALLED BY IS USED BY	2C8 - AJACOB - AJACOB MAIN	INSTR ANAL MANU	JF6CIN CONST MANU	WHEN CONTR BNGKTA	MTLT DERIV ROTAN	QUAN FOCUS RTINIT	SHSAA HAMS STAB	MPCC START	IMMO TRIM	INSTAB TVT	ITRIM VAR	JACOB
MOAL	LENGTH 450 CALLED BY - ITRCT IS USED BY - AJACOB MANU CALLS - ANDOIT		ANAL MANU MANAL	CONST BNGKTA STAB	CONTR ROTAN TOPLOT	DERIV STAB	FOCUS TRIM	HAMS TVT	MPCC	INSTAB	ITRIM	JACOB	MAN
MONDRS	LENGTH 788 CALLED BY - MCOES IS USED BY - CONST CALLS - MANAL		MAN STAB	STBO	STRIAB	STRIAB							
MREN	LENGTH 800 CALLED BY - START IS USED BY - MAIN CALLS - FCAY USES - CGZARH STAB		FORVY STRIAB FLEX STAB	FORVY SMAS FORVY STAB	INSTAB TILT FORME STRIAB	MANAL TOPLOT PORT STRIAB	MATRIX TURN FORVY TAB	MOPS VMO FORVY TAB	RESTRY	SAVOLD	STVAR	STAB	STAB
MODAL	LENGTH 848 CALLED BY - IMBO IS USED BY - MAIN CALLS - FLEX USES - FLEX		RTINIT INSTAB INSTAB	START MANAL MANAL	STAB STAB STAB	STAB STAB	STAB STAB	STRIAB	UNOAL				
MODES	LENGTH 938 CALLED BY - CONST IS USED BY - MAIN CALLS - IDIAT USES - DATE		MANAL INSTAB	MODLS MANAL	PUNCH STAB	STAB STAB	STBO STBO	STRIAB STRIAB	STRIAB STRIAB	TOPLOT			
MONB	LENGTH 878 CALLED BY - VARI IS USED BY - DERIV CALLS - FCAY USES - CGZARH		HAMS FORVY INSTAB	MPCC FORVY MANAL	MAIN MANAL STAB	MANU MOPS STRIAB	MANU SOLVE TOPLOT	BNGKTA STAB ZLLCAL		TI .Y	TOPLOT		
MTLT	LENGTH 848 CALLED BY - VARI IS USED BY - DERIV CALLS - MANAL USES - CGZARH		HAMS MATRIX INSTAB	MPCC STAB MANAL	MAIN STRIAB STAB	MANU TILT STRIAB	MANU TOPLOT TOPLOT	BNGKTA STAB ZLLCAL					
MOPS	LENGTH 508 CALLED BY - WHEN IS USED BY - CONTR CALLS - MANAL		MONB DERIV	TRIM HAMS	MPCC	MAN	MANU	MANU	BNGKTA	START	TRIM	VAR	

TABLE 4-1. Continued.

INPUT	LENGTH CALLED BY - READIN IS USED BY - MAIN CALLS - ERRORS USES - INSTAR	START INSTAR TOPLOT	MANAL	STAMAN	STARAD	STRINA	MPCC TVT	ITRIM	ITRIB
INSTED	LENGTH 776 CALLED BY - RADIAL IS USED BY - AJACOB CALLS - ANOIT USES - ANOIT	ANAL MAIN DIFFER	AZMUTH MANAL STARAD	CONSTB MANAL STARAD	CORTRN RICKTA STARAD	DERIV ROTAN	FOCUS STAB	ITRIM	ITRIB
POSRED	LENGTH 216 CALLED BY - REDID IS USED BY - JSTED	MAIN	READIN	VEDATB	REDONE	REDONE	START		
PBZ1	LENGTH 388 CALLED BY - ITRIM IS USED BY - CONTRM	TRIM MAIN	TRIM						
PUNCH	LENGTH 660 CALLED BY - MCODES IS USED BY - CONSTB CALLS - DATE	MAIN INSTAR	STAMAN	STOB	STRINA	TOPLOT			
PYLINT	LENGTH 616 CALLED BY - INNO IS USED BY - MAIN CALLS - FLEX	RTINJT INSTAR	START MANAL	STAMAN	STARAD	STARAD			
QUAN	LENGTH 476 CALLED BY - DEPIV IS USED BY - MPMPS CALLS - FLEX	MANAL MPCC FORV	MANAL MAIN FORVYV	MANAL FORVYV	MANAL MANAL	RICKTA MATRIX	STARAD		
RADGCH	LENGTH 920 CALLED BY - RADIAL IS USED BY - AJACOB CALLS - ANOIT USES - ANOIT	ANAL MAIN FLEX DOTX	AZMUTH MANAL FORUM FORUM	CONSTB MANAL FORUM1 FORUM1	CORTRN RICKTA MANAL MANAL	DERIV ROTAN RICKTA STARAD	FOCUS STAB RTWAKE STARAD	ITRIM	ITRIB
RADIAL	LENGTH 736 CALLED BY - AZMUTH IS USED BY - AJACOB CALLS - ANOIT USES - ANOIT	ANAL MAIN CDCL UNSTED CDCL TAB	CONSTB MANAL CDCL DIFFER TABINT	CORTRN RICKTA FLEX DOTX TAB1	DERIV ROTAN MANAL FLEX TOPLOT	FOCUS STAB RICKTA STARAD	MPCC TVT STARAD	ITRIM	ITRIB
RADOUT	LENGTH 676 CALLED BY - AZMUTH IS USED BY - AJACOB CALLS - ANOIT USES - ANOIT	ANAL MAIN CDCL UNSTED CDCL TAB	CONSTB MANAL CDCL DIFFER TABINT	CORTRN RICKTA FLEX DOTX TAB1	DERIV ROTAN MANAL FLEX TOPLOT	FOCUS STAB RICKTA STARAD	MPCC TVT STARAD	ITRIM	ITRIB

TABLE 4-1. Continued.

[illegible]

TABLE 4-1. Continued.

RMKTA	448 CALLED BY - MPCC IS USED BY - MAIN CALLS - DERIV USES - ANAL FLEX MCCSP RADONH STBZFN UNSTED	MANUV FORVYV ANDOTT FOCUS INSTAR RADIAL STRIAB VARI	FORVYV AZMUTH FORCNC INTPRO RADOUT STRIAB VCNTRL	MANAL AZMUTH FORCNC ITROT RGUST SUPERP VIND	STRINA COCL FORMK1 MANAL ROTAN SNAS MAG	CGZARH FORV MATRIX RTWAKE SVSRAT WING	CLCD FORVYV MHAL SCASIT TAB WSDOOF	CHCALC FORVYV MOMB TABINT XSTORE	CHTN FORMK1 MTLY STANAM TAB1 ZLLCAL	DIFFER FORMK1 MOP'S STARAD TILT	DOTX FUSPMH NSTED STARAD TOPLOT	EXTOAS GUST QUAN STBNAX TVT
ROTAN	440 CALLED BY - ANAL IS USED BY - AJACOB CALLS - RMKTA USES - ANDOTT FOCUS AZMUTH FORVYV FCRY RADOUT UNSTED	DERIV CONSTB STAB FOCUS AZMUTH FORVYV RGUST VIND	TRIN CONTRM TRIN	DERIV STARAD COCL WRESP STARAD	MANMS TOPLOT CHCALC INTPRO STARAD	MPCC TVT DIFFER ITROT STARAD	INSTAB	ITRIN	JACOBI	MAIN	MANU	MANUV
RPTPG	1448 CALLED BY - TRIP IS USED BY - CONTRM CALLS - ANDOTT USES - DATE	MAIN FLEX TOPLOT	INSTAR	MANAL	STARAD	STARAD	STARAD	STARAD	STRINA	WROTI		
RTBRT	340 CALLED BY - START IS USED BY - MAIN CALLS - INRO USES - FLEX TOPLOT	INSTAR INBOLD WRODAL	MANAL INMTR	STARAD INSTAR	STARAD MANAL	STARAD MATRIX	STRAB MODAL	STRAB PYLIN	TOPLOT STARAD	STARAD	STARAD	STRINA
RTWAKE	640 CALLED BY - RANPCN IS USED BY - AJACOB CALLS - JACOBI USES - DOTX	ANAL MAIN FORMK1	AZMUTH MANU FORMK1	CONSTB MANUV MANAL	CONTRM RADIAL	DERIV RMKTA	FOCUS ROTAN	MANMS STAB	MPCC TRIM	INSTAB TVT	ITRIN	ITROT
SAVEIC	408 CALLED BY - MPCC IS USED BY - MAIN CALLS - FORVYV USES - FORVYV	MANUV FORVYV FORVYV	MANAL MANAL	SAVOLD STARAD	STARAD STRINA	STRINA						
SAVOLD	368 CALLED BY - MPCC IS USED BY - MANMS CALLS - FORVYV	INIT MPCC FORVYV	WHEN MAIN MANAL	SAVEIC MANU STARAD	MANUV STRINA	START	TIMEP					
SCASIT	618 CALLED BY - DERIV IS USED BY - MANMS CALLS - FORV	MPCC FORVYV	MAIN FORVYV	MANU MANAL	MANUV STARAD	RMKTA STRINA	TOPLOT					

TABLE 4-1. Continued.

[illegible]

TABLE 4-1. Continued.

[illegible]

SUPERP	LENGTH 840 CALLED BY - VARI IS USED BY - DERIV CALLS - MANAL USES - STRINA	HMM'S SOLVE TOPLOT	MPCC STANAM	MAIN STRINA	NAMU TOPLOT	NAMUV	RNGKTA
SWAS	LENGTH 670 CALLED BY - AJACOB IS USED BY - CONSTB START CALLS - MANAL	MHEM CONTRM TRIM STAPAM	STAB DERIV STRINA	VARI HMM'S	MPCC	INSTAB	JACOBI ITRIM MAIN NAMU NAMUV RNGKTA
SUBRAT	LENGTH 740 CALLED BY - FOCUS IS USED BY - AJACOB NAMUV CALLS - AMOITY	ANAL RNGKTA FLEX	CONSTRB ROTAN FORUMI	CONTRM STAB	DERIV TRIM MANAL	HMM'S TVT MATRIX	MPCC JACOBI ITRIM MAIN NAMU NAMUV RNGKTA
TAB	LENGTH 630 CALLED BY - REDATO IS USED BY - AJACOB INSTAB ROTAN	RESTRY ANAL ITRIM STAB	TABINT COCL JACOBI STBZFN	CONTRM STAB	DERIV TRIM MANAL	HMM'S TVT MATRIX	MPCC JACOBI ITRIM MAIN NAMU NAMUV RNGKTA
TABPIX	LENGTH 308 CALLED BY - TABOUT IS USED BY - MAIN	START	TABINT COCL JACOBI STBZFN	CONTRM STAB	DERIV TRIM MANAL	HMM'S TVT MATRIX	MPCC JACOBI ITRIM MAIN NAMU NAMUV RNGKTA
TABINT	LENGTH 898 CALLED BY - COCL IS USED BY - AJACOB INSTAB WING	GLCD ANAL MAIN	AZMUTH NAMU	CONSTB NAMUV	CONTRM RADIAL	DERIV RNGKTA	FOCUS MPCC INSTAB ITRIM ITROT UNSTED
TABOUT	LENGTH 740 CALLED BY - START IS USED BY - PAIR CALLS - INSTAR	TAB	TABPIX	TAB1	CONTRM RADIAL	DERIV RNGKTA	FOCUS MPCC INSTAB ITRIM ITROT UNSTED
TAB1	LENGTH 5400 CALLED BY - REDATO IS USED BY - AJACOB INSTAB ROTAN	RESTRY ANAL ITRIM STAB	TABINT AZMUTH ITROT START	TABOUT COCL JACOBI STBZFN	GLCD JSTRED TIMLP	CONSTB MAIN TRIM	CONTRM NAMU TVT FOCUS MPCC INSTAB ITRIM ITROT UNSTED
TYLT	LENGTH 508 CALLED BY - MHEM IS USED BY - DERIV CALLS - CELEARN USES - INSTAR	MHEM MPCC STANAM	MTLT MPCC STANAM	VARI MAIN STRINA	NAMU TOPLOT	NAMUV ZLLCAL	START BNGKTA VARI

TABLE 4-1. Continued.

[illegible]

TABLE 4-1. Continued.

TVT	LENGTH 1308 CALLED BY - ROTAN IS USED BY - AJACOB NAMUV CALLS - AMOIT USES - AMOIT FORVYV RGUST VINO	ANAL RNGKTA FLEX AZMUTH FORVYV RTWAKE	CONSTB STAR FOCUS AZMUTH HRESP STARAN	CONTRM TRIM FORV CDCL INT-RO STARAD	DERIV NAMAL CHCALC ITROT STARAN	NAMMS STARAN DIFFER NAMAL STRINA	MPCC STARAN DOTX MATRIX SHSRAT	INSTAB TOPLOT FLEX MBAL TAB	ITRIM FORCNC NSTED TABINT	JACOB I FORCNC RADACH TAB I	MAIN FORCNC RADIAL TOPLOT	NAMU FORV RADOUT UNSTED
UNSTED	LENGTH C88 CALLED BY - RADIAL IS USED BY - AJACOB JACOB I CALLS - AMOIT USES - AMOIT	ANAL MAIN CDCL NAMAL	AZMUTH NAMU NAMAL STARAD	CONSTB PIAMUV STARAD	CONTRM RNGKTA STARAN TAB	DERIV ROTAN TOPLOT TABINT	FOCUS STAR TAB I	NAMMS TRIM	MPCC TVT	INSTAB	ITRIM	ITROT
VARI	LENGTH E08 CALLED BY - DERIV IS USED BY - NAMMS CALLS - CNTH SUPERP USES - CGJARM STRIB	MPCC EXTORS SHAS FLEX STRIPA	MAIN FLEX TILT FORV TILT	NAMU FORV TOPLOT FORVYV TOPLOT	NAMUV FORVYV VCNTRM FORVYV ZLLCAL	RNGKTA FORVYV ZLLCAL INSTAR	GUST NAMAL	NAMAL MATRIX	NOMB NOPS	MTLT SOLVE	STARAN STARAN	STRINA STARAN
VCNTRM	LENGTH D60 CALLED BY - VARI IS USED BY - DERIV CALLS - FLEX	NAMMS FORV	MPCC FORVYV	MAIN FORVYV	NAMU NAMAL	NAMUV STARAN	RNGKTA STRINA	TOPLOT				
VINO	LENGTH 438 CALLED BY - ITROT IS USED BY - AJACOB NAMU CALLS - NAMAL	PMER ANAL NAMUV STARAD	CONSTB RNGKTA STARAN	CONTRM ROTAN	DERIV STAR	FOCUS START	NAMMS TRIM	MPCC TVT	INSTAB	ITRIM	JACOB I	MAIN
VAC	LENGTH 600 CALLED BY - VINC IS USED BY - AJACOB NAMUV CALLS - STARAN	ANAL RNGKTA	CONSTB STAR	CONTRM TRIM	DERIV	NAMMS	MPCC	INSTAB	ITRIM	JACOB I	MAIN	NAMU
VTIB	LENGTH C70 CALLED BY - ANAL IS USED BY - AJACOB RNGKTA CALLS - AMOIT USES - AMOIT TOPLOT	CONSTB STAR CLCD DOTX	CONTRM TRIM FORMEL FORMEL	DERIV FORMEL FORMEL	NAMMS NAMAL FORMEL	MPCC STARAN FORMEL	INSTAB STARAD NAMAL	ITRIM STARAN STARAN	JACOB I STARAN STARAN	MAIN STRINA TAB	NAMU TOPLOT TABINT	NAMUV MAG TAB I
VRTAM	LENGTH 520 CALLED BY - REDRUK IS USED BY - JSTRED	MAIN	READIN	START								

TABLE 4-1. Continued.

WRPH	LENGTH 450 CALLED BY - ALJACOB IS USED BY - CONSTB CALLS - MANUAL	WRINST CONTRM TRIM STRINA	WRNAMU NAMMS	WRTRIM MPCC	INIT	INSTAB	ITRIM	JACOBI	MAIN	NUMU	NUMUV	STAB
WRINST	LENGTH 1908 CALLED BY - INSTAB IS USED BY - CONSTB CALLS - MANUAL USES - MANUAL	STAB NAMM STANAM STRINA	STBD	STRAB	STRINA	WRPH						
WRNAMU	LENGTH DAO CALLED BY - INIT IS USED BY - NAMMS CALLS - DATE USES - MANUAL	MPCC FORV STANAM	MAIN FORVYI STRINA	NUMU FORVYI TOPLOT	NUMU MANAL	TRIM STANAM	STRINA	TOPLOT	WRPH	WRTHUV		
WRBDAL	LENGTH 1068 CALLED BY - MICAL IS USED BY - INRO CALLS - FLEX	MAIN INSTAR	RTINIT MANAL	START STANAM	STANAM	STRINA						
WRPHS	LENGTH CPO CALLED BY - ALSTAB IS USED BY - CONSTB CALLS - STANAM	MAIN STBD										
WRPHI	LENGTH 1P8 CALLED BY - MAIN IS USED BY - CONTRM CALLS - DATE	READIN MAIN TOPLOT	RPTPG START	WRTRIM TRIM								
WRSTAB	LENGTH 1108 CALLED BY - STAB IS USED BY - CONSTB CALLS - STBD	MAIN										
WRTHUV	LENGTH 1230 CALLED BY - WRNAMU IS USED BY - CONTRM CALLS - MANUAL	WRTRIM NAMMS STANAM	MPCC STRINA	INIT TOPLOT	MAIN	NUMU	NUMUV	TRIM				
WRTRIM	LENGTH 488 CALLED BY - TRIM IS USED BY - CONTRM CALLS - DATE USES - DATE	MAIN MANAL FORV	STABAD FORVYI	STANAM FORVYI	STRAB INSTAR	STRINA MANAL	TOPLOT STANAM	THANAM STABAD	WRPH	WRTHUV STRINA	WRTHUV TOPLOT	
WRVP	LENGTH 640 CALLED BY - ALJACOB IS USED BY - CONSTB CALLS - MANUAL	ITRIM CONTRM STRINA	WRTRIM INSTAB STRINA	ITRIM	JACOBI	MAIN	TRIM					

TABLE 4-1. Concluded.

WSPWOF	LENGTH 2AB CALLED BY - FUSWOF IS USED BY - AJACOB CALLS - STABAN	ANAL RNGKTA STRINA	CONSTB STAB	CONTRM TRIM	DERIV	NAMES	NPCC	INSTAB	ITRIM	JACOBI	MAIN	NAMU	NAMU
WCONIN	LENGTH 5CB CALLED BY - START IS USED BY - MAIN CALLS - INSTAB	NAMAL	STABAN	STRINA	TOPLOT								
WSTINT	LENGTH 2DB CALLED BY - START IS USED BY - MAIN CALLS - INSTAB	STRINA											
WSTCRE	LENGTH 59B CALLED BY - ANAL IS USED BY - AJACOB CALLS - NAMAL	CONSTB STAB STRINA	CONTRM TRIM	DERIV	NAMES	NPCC	INSTAB	ITRIM	JACOBI	MAIN	NAMU	NAMU	NAMU
WVBWIT	LENGTH 8B0 CALLED BY - START IS USED BY - MAIN												
WVBWIT	LENGTH 85B CALLED BY - START IS USED BY - MAIN												
ZEND	LENGTH 5E0 CALLED BY - LIZE IS USED BY - MAIN CALLS - FLEX	START FORMK	FORMK1	NAMAL	STABAN	STRAB	STRINA						
ZLICAL	LENGTH 3AB CALLED BY - AJACOB IS USED BY - CONSTB CALLS - NAMAL	STAB CONTRM HTLT STABAN	TILT DERIV RNGKTA STRINA	VARI NAMES START	NPCC TRIM	INSTAB VARI		JACOBI	MAIN	NAMU	NAMU	NAMU	NAMU

**TABLE 4-2. CROSS-REFERENCE OF MULTIPLE ENTRY
SUBROUTINES IN AGAJ73**

Name of Additional Entry Point	Contained in Subroutine	Name of Additional Entry Point	Contained in Subroutine
AJACB1	AJACOB	RESTO	RESTR1
ALLVEC	ALLMAT	REST1	RESTR1
ATMINT	JFBGIN	REST2	RESTR1
ATRM	TRIM	REST4	RESTR1
AUXJ	CNTM	RWKOUT	REDRWK
AZMOUT	RADOUT	SUBA	MOMB
CGXARM	CGZARM	SUBB	MOMB
CGYARM	CGZARM	SWKOUT	REDSWK
FLAP	CNTM	TFFA	TILT
FLAT	MTLT	TTLT	MTLT
HARM1	HARM	VCNT1	VCNTRL
HSAF	TILT	VCNT2	VCNTRL
INBLD1	INBLD	VCNT3	VCNTRL
INBLD2	INBLD	VCNT4	VCNTRL
INBLD3	INBLD	VCNT5	VCNTRL
INBLD4	INBLD	VCNT6	VCNTRL
IOWRFM	WRINST	VCNT7	VCNTRL
IOWRF1	WRINST	VCNT8	VCNTRL
NOPS1	NOPS	VCNT9	VCNTRL
NSTED1	NSTED	VCNT10	VCNTRL
PDZ	PDZ1	WRMDL1	WRMDAL
RATI	DAMPER	WRMDL2	WRMDAL
RESTOR	SAVEIC	WROT	WROT1
RESTR1	SAVEIC	YAWP	MTLT

TABLE 4-3. CONTROL SECTION CROSS-REFERENCE FOR GDAJ07

PURG IHCECOMH	CONTROL	SECTION	REMOVED
PURG IHCUOPT	CONTROL	SECTION	REMOVED
PURG IHCCOMH2	CONTROL	SECTION	REMOVED
PURG IHCEERM	CONTROL	SECTION	REMOVED
PURG IHCLSORT	CONTROL	SECTION	REMOVED
PURG IHCFVTH	CONTROL	SECTION	REMOVED
PURG IHCSORT	CONTROL	SECTION	REMOVED
PURG IHCEFIOS	CONTROL	SECTION	REMOVED
PURG IHCUATBL	CONTROL	SECTION	REMOVED
PURG IHCEFNTH	CONTROL	SECTION	REMOVED
PURG IHCSSEN	CONTROL	SECTION	REMOVED
PURG IHCETRCH	CONTROL	SECTION	REMOVED
PURG IHCLATN2	CONTROL	SECTION	REMOVED
PURG IHCSLNG	CONTROL	SECTION	REMOVED
PURG IHCESEXP	CONTROL	SECTION	REMOVED
PURG IHCFRXP	CONTROL	SECTION	REMOVED
PURG IHCSATN2	CONTROL	SECTION	REMOVED
PURG IHCIERH	CONTROL	SECTION	REMOVED
PURG IHCFRXP	CONTROL	SECTION	REMOVED
PURG IHCLSCN	CONTROL	SECTION	REMOVED

SPRIV001	LENGTH	30										
SPRIV002	LENGTH	60										
AXIS#	LENGTH	800										
	IS USED BY -	COMPLT	FSFT	MAIN								
	CALLS -	CALCON	NUMBER	SYMBOL								
	USES -	CALCON	NEXTTIME	SYMBOL								
CALC'M	LENGTH	800	CALC01	COMPLT	LINE	MAIN	PLOT#	SCALIT	SYMBOL	SCALIT		
	IS USED BY -	AXIS#	CALC01	COMPLT	FSFT	LINE	MAIN					
CALC91	LENGTH	1498			LINE	NUMBER	SYMBOL	TOPLOT				
	IS USED BY -	COMPLT			LINE	NUMBER	SYMBOL	TOPLOT				
	CALLS -	CALCON			LINE	NUMBER	SYMBOL	TOPLOT				
	USES -	CALCON			LINE	NUMBER	SYMBOL	TOPLOT				
COMPLT	LENGTH	148	CURVET	CALCON	PLOT#	SCALIT	MAIN	SCALIT	LINE	MAXMIN	PLOT#	
	IS USED BY -	COMPLT	CURVET	CALCON	PLOT#	SCALIT	MAIN	SCALIT	LINE	MAXMIN	PLOT#	
	CALLS -	CALCON	CURVET	CALCON	PLOT#	SCALIT	MAIN	SCALIT	LINE	MAXMIN	PLOT#	
	USES -	CALCON	CURVET	CALCON	PLOT#	SCALIT	MAIN	SCALIT	LINE	MAXMIN	PLOT#	
CURVET	LENGTH	1090	TIMPTS	PLOT#	TOPLOT	WROT1	TOPLOT					
	IS USED BY -	COMPLT	TIMPTS	PLOT#	TOPLOT	WROT1	TOPLOT					
	CALLS -	WEDING	TIMPTS	PLOT#	TOPLOT	WROT1	TOPLOT					
	USES -	DATE	TIMPTS	PLOT#	TOPLOT	WROT1	TOPLOT					
COIL	LENGTH	488			TOPLOT							
	IS USED BY -	COMPLT			TOPLOT							
	CALLS -	WEDING			TOPLOT							
	USES -	DATE			TOPLOT							
DATE	LENGTH	1F0	CURVET	MAIN	PLOT#	SCALIT						
	IS USED BY -	WROT1	CURVET	MAIN	PLOT#	SCALIT						
	CALLS -	COMPLT	CURVET	MAIN	PLOT#	SCALIT						
FSFT	LENGTH	11RC0	HEDING	CALCON	TOPLOT	NEXTTIME	NUMBER	PLOT#	SCALE#	SYMBOL		
	IS USED BY -	COMPLT	HEDING	CALCON	TOPLOT	NEXTTIME	NUMBER	PLOT#	SCALE#	SYMBOL		
	CALLS -	WEDING	HEDING	CALCON	TOPLOT	NEXTTIME	NUMBER	PLOT#	SCALE#	SYMBOL		
	USES -	AXIS#	HEDING	CALCON	TOPLOT	NEXTTIME	NUMBER	PLOT#	SCALE#	SYMBOL		

TABLE 4-3. Continued.

NAME	LENGTH CALLED BY IS USED BY	380 - FSFT - COMPT	MAIN	DATE PPLOT	FSFT SCALE#	HARM SCALIT	HEDING SCLFIX	IMPLT SYMBOL	LINE TIMPTS	MAXMIN TOPLOT
HEDING	LENGTH 840 CALLED BY - CALC01 IS USED BY - COMPT CALLS - PLOT01		CURVET MAIN PLOT01							
IMPLT	LENGTH 1900 CALLED BY - CALC01 IS USED BY - COMPT	THIS IS A 'COMMON' CONTROL SECTION PLOT SCALIT SCLFIX MAIN SCALIT								
LINE	LENGTH 448 CALLED BY - CALC01 IS USED BY - COMPT CALLS - CALCOM USES - CALCOM		PLOT01 FSFT SYMBOL NEXTTIME							
MAIN	LENGTH 1438 CALLS - CALCOM USES - AXIS# NEXTTIME WROT1	COMPT CALCOM NUMBER	MAXMIN CALC01 PLOT01	WROT1 CURVET PLOT01						
MAXMIN	LENGTH 2800 CALLED BY - COIL IS USED BY - COMPT	THIS IS A 'COMMON' CONTROL SECTION MAIN SCALIT MAIN SCALIT								
NEXTTIME	LENGTH 8 CALLED BY - SYMBOL IS USED BY - AXIS#	CALC01	COMPT	FSFT	LINE	MAIN				
NUMBER	LENGTH 170 CALLED BY - AXIS# IS USED BY - COMPT CALLS - SYMBOL USES - CALCOM	CALC01 FSFT	PLOT01 MAIN	PLOT01 SCALIT						
PLOT01	LENGTH 1468 CALLED BY - HEDING IS USED BY - CALC01	COMPT	CURVET	FSFT	MAIN	PPLOT	SCALIT			
PLOT01	LENGTH 442 CALLED BY - HEDING IS USED BY - CALC01	COMPT	CURVET	FSFT	MAIN	PPLOT	SCALIT			
PLOT01	LENGTH 478 CALLED BY - FSFT IS USED BY - COMPT CALLS - AXIS# USES - CALCOM	MAIN CALCOM NEXTTIME	LINE NUMBER SYMBOL	SCALES#						

PLOT	LENGTH CALLED BY - SCALIT IS USED BY - COMPLY CALLS - MEDING USES - DATE	MAIN IMPLT PLOT	WROT1 PLOT1	TOPLOT
SCALE0	LENGTH 510 CALLED BY - PLOT IS USED BY - COMPLY FSFT	MAIN		
SCALIT	LENGTH 10740 CALLED BY - COMPLY IS USED BY - MAIN CALLS - CALCOM USES - CALCOM WROT1	IMPLT MEDING	SCALFX LINE	TOPLOT MAXMIN
SCALFX	LENGTH 430 CALLED BY - SCALIT IS USED BY - COMPLY CALLS - IMPLT	MAIN MAXMIN	TOPLOT	
SYMBOL	LENGTH 809 CALLED BY - AXIS IS USED BY - AXIS CALLS - CALCOM NEXTTIME	CALC01 LINE CALC01 COMPLY NEXTTIME	PLOT NUMBER FSFT	SCALIT
TIMPTS	LENGTH 2030 THIS IS A 'COMMON' CONTROL SECTION CALLED BY - CURVET IS USED BY - COMPLY COIL MAIN			
TOPLOT	LENGTH 108 THIS IS A 'COMMON' CONTROL SECTION CALLED BY - CALC01 IS USED BY - COMPLY CURVET MAIN	MAIN PLOT SCALIT	SCALIT SCALFX WROT1	
WROT1	LENGTH 203 CALLED BY - CURVET IS USED BY - COMPLY CALLS - DATE	MAIN PLOT SCALIT		

TABLE 4-4. REDUCTION OF AGAJ73 STORAGE REQUIREMENTS									
Array Name Outside RESTRT		COMMON Name	Dimensions of Array				Approx. Reduction of Storage Reqmts.**		
			Outside RESTRT		In RESTRT				
			Standard	Reduced	Standard	Reduced	Standard	Reduced	Bytes(x10 ⁻³)
WKTR	FRK1	{ TAB1 }	(33,10,5,20,2)	(1,1,1,1,1)	(66000)	(1)	264	260	
WKSTB	FSK1		(15,10,5,12)	(1,1,1,1,1)	(9000)	(1)	36	34	
CURVED	TABL		(1100,5)*	(1100,2)*	(6350)	(3050)	13	12	
CURVEL	{ TAB1 }		(500,5)*	(500,2)*	{ (5375) }	{ (2150) }	6	6	
CURVEM		(575,5)*	(575,2)*	7			6		
AUX	PHY1		(16,215)	(1,1)	(3440)	(1)	14	12	
*In block data DAT1, the second subscript of the standard array is 4 rather than 5 and that of the reduced array is 1 rather than 2; the remaining entries of the COMMON block are filled with the block data (e.g., CL5A, CL5B, etc., for TAB). The block data in DAT1 are the data tables for an NACA 0012 airfoil. Also, if TAB and TAB1 are reduced, three statements near the end of subroutine TABOUT must be changed. The location of each statement is given with respect to the last call to subroutine TABFIX in TABOUT (i.e., the nth statement after the call).									
Location		Standard			Reduced				
1 st 5 th 7 th		IF (K.EQ.5) RETURN			IF (K.EQ.2) RETURN				
		K=5			K=2				
		IF (YRR(18,1).EQ.5.) .OR. YSTBZ(18,1).EQ.5.) GO TO 26			IF (YRR(18,1).EQ.2.) .OR. YSTBZ(18,1).EQ.2) GO TO 26				
**1K = 1024 bytes									

TABLE 4-5. STAB DIAGNOSTIC SWITCH IN AGAJ73

IPL(34)	Variable
1	FUS. U
2	FUS. W
3	FUS. Q
4	FUS. V
5	FUS. P
6	FUS. R
7	M.R. F/A PYLON RATE
8	M.R. LAT PYLON RATE
9	T.R. F/A PYLON RATE
10	T.R. LAT PYLON RATE
11	M.R. F/A FLAP. RATE
12	M.R. LAT FLAP. RATE
13	T.R. F/A FLAP. RATE
14	T.R. LAT FLAP. RATE
15	M.R. F/A PYLON DISP
16	M.R. LAT PYLON DISP
17	T.R. F/A PYLON DISP
18	T.R. LAT PYLON DISP
19	M.R. F/A FLAP. DISP
20	M.R. LAT FLAP. DISP
21	T.R. F/A FLAP. DISP
22	T.R. LAT FLAP. DISP

```

//E2410480 JOB (AS812A00.G38.69109308.DP20.Y.01)..'PY
//          MSGLEVEL=1.CLASS=C
//JOBLIB   DD DSN=ENGTFST
//GO EXEC  PGM=AS812A
//SYSPRINT DD SYSOUT=A
//CARDS    DD SYSOUT=B.DCB=(LRECL=80,BLKSIZE=80,RECFM=F)
//SYSIN    DD *
/*

```

2841'..*

Figure 4-1. Job Control Language To Run AS812A.